CASE STUDIES OF OPTIMUM FILTER-CONTROLLER DESIGN IN SAMPLED DATA SYSTEMS

JOHN COLEMAN McMICHAEL
and
GEORGE CHRIST SKEZAS

CASE STUDIES OF OPTIMUM FILTER-CONTROLLER DESIGN IN SAMPLED DATA SYSTEMS

by

John Coleman McMichael
Lieutenant, United States Navy
B. S., United States Naval Academy, 1958

and

George Christ Skezas
Lieutenant, United States Navy
B. S., United States Naval Academy, 1958

Submitted in partial fulfillment
for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

UNITED STATES NAVAL POSTGRADUATE SCHOOL
May 1966

ABSTRACT

In this paper an investigation is made of the problem of estimating and predicting the states of a linear, discrete, time-invariant, dynamic process which is excited by Gaussian noise and where the observable states are disturbed by Gaussian measurement noise. The concepts of optimum filter design, originally developed by R. E. Kalman, are utilized. Also we have closed the loop on two illustrative examples by determining the optimal control for the plant as a function of the plant's state variables. Here the concepts of a cost performance index and dynamic programming (the latter originally developed by R. Bellman) are employed.

The CDC 1604 digital computer, using Fortran 60 programming is utilized in the solution of the optimum filter-controller design.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. Introduction.

In this paper many of the loose ends to the sampled-data optimal control problem are drawn together. The complete control problem is consolidated in the design of an optimal filter and an optimal feedback controller for a system contaminated by random exitation and measurement noise. Further, by incorporating deterministic inputs, we are capable of closing the loop on the entire system.

The first several sections offer background information and discuss the actual design of the optimal filter and optimal controller. We then illustrate with two examples the power of the methods discussed. Each example represents a real system, and the results indicate that any order system may be controlled successfully as outlined.

2. Random Noise Processes.

Because the design of the optimal filter is based upon the concepts of probability and the statistics of a random process (noise in this case), it is felt that a review of these concepts is in order. Since we are dealing with noise, a random process, we are not specifically interested in a system's response to a single, isolated noise signal, but rather its expected response over the entire range of noisy signals.

A random variable is a function whose values depend on the outcome of a chance event. Thus a random variable is unpredictable in assuming its different possible values. We will consider random variables whose values are real numbers. A random variable, x, must have a set of possible

values and a probability associated with each value. The mathematical behavior of a random variable can be expressed in terms of its distribution function, which gives a specification of the possible values a random variable may assume together with their respective probabilities. The probability that x is less than or equal to a, is denoted by the symbol $F_x(a)$, thus:

$$P(x \leq a) = F_x(a) \tag{2.1}$$

where $F_x(a)$ is called the distribution function.

If $F_x(a)$ is differentiable with respect to a, then:

$$f_x(a) = \frac{dF_x(a)}{da} = \text{probability density function of } x$$

The expected value (or mean or statistical average) of a random variable x, is:

$$E(x) = \int_{-\infty}^{+\infty} x\, f_x(a)\, da \quad \text{for a continuous random variable} \tag{2.2a}$$

$$= \sum_{i=1}^{N} x_i\, f_{x_i}(a) \quad \text{for a discrete random variable} \tag{2.2b}$$

Thus the expected value or mean is an indication of the location of the center of gravity. Now that the concept of the mean has been established, we are concerned with the concentration of the random variable about the mean. The simplest measure of this concentration is the mean square deviation from the norm. When the mean, μ, is chosen as the norm, the measure of the departure from the norm (dispersion) is called the variance.

$$\text{Variance} = \sigma^2 = E\left[(x - \mu)^2\right] \tag{2.3}$$

$$\text{Standard Deviation} = \sigma = \sqrt{\sigma^2} \tag{2.4}$$

It can further be shown that:

$$\sigma^2 = E(x^2) - \mu^2 \tag{2.5}$$

Thus knowledge of a random variable's mean and standard deviation will describe the distribution of the values of the elements of the process.

Now consider the most common type of noise process, a GAUSSIAN PROCESS, which has the probability density function:

$$p(x) = f_x(a) = \frac{1}{\sqrt{2\pi\sigma}} \exp \frac{x^2}{2\sigma^2} \tag{2.6}$$

This process has a mean of zero and a variance of σ^2. The probability density is independent of time and therefore called a STATIONARY PROCESS. Figure (2-1) shows the probability density function for a Gaussian process.

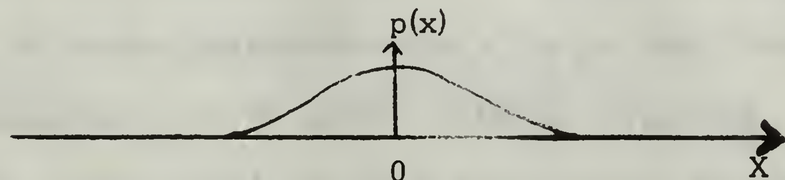

Figure 2-1 Gaussian Probability Density Function

It is noted that most of the elements of the process have values in the neighborhood of the origin.

A useful and important concept is that of the joint probability density function of values of the process at two different times. The second-order probability density function of a Gaussian noise process can be defined as:

$$\frac{1}{2\pi\sigma^2 \sqrt{1-\rho^2}} \exp \left[\frac{-x_1^2 + 2\rho x_1 x_2 - x_2^2}{2\sigma^2 (1-\rho^2)} \right] \tag{2.7}$$

In this expression x_1 and x_2 are the values of the noise process at times t_1 and t_2 and ρ is a factor indicating the degree of correlation between x_1 and x_2. When t_1 and t_2 are close together ($t_2 - t_1 = \tau \longrightarrow 0$) so that x_1 and x_2 have about the same values, then the value of ρ is close to 1 indicating that there is a high degree of correlation.

The factor ρ, mentioned above is known as the normalized autocorrelation function. We shall be interested, however, in the average value of the product $x_1 . x_2$ (denoted by $\overline{x_1 . x_2}$) which is known as the autocorrelation function and, in a stationary process, is a function of the time difference $t_2 - t_1 = \tau$. Thus:

$$\phi(\tau) = \overline{x_1 . x_2} \qquad (2.8)$$

When τ is very large, x_1 and x_2 are uncorrelated, except for the average value, thus, $\phi(\pm \infty) = \left[E(x)\right]^2$. When $\tau = 0$, the autocorrelation function equals the average value of x^2, or $\phi(0) = E(x^2)$. Thus the variance of the process can be given by:

$$\sigma^2 = \phi(0) - \phi(\infty) \qquad (2.9)$$

For the Gaussian process the mean, μ or $E(x)$, is zero, hence $E(x)^2$ is zero, therefore:

$$\sigma^2 = \phi(0) \qquad (2.10)$$

The Concept of Power Density and the Autocorrelation Function

Fourier analysis yields the frequency spectrum of a function as follows:

$$X(\omega) = \int_{-T}^{T} x(t) \ e^{-j\omega t} \, dt \tag{2.11}$$

In order to obtain the power spectra, we must divide the energy spectra by the observation time, and it can be shown:

$$N(\omega) = \frac{1}{2T} \left| X(\omega) \right|^2 = \frac{1}{2T} \int_{-T}^{T} e^{-j\omega\tau} \, d\tau \int_R x(t+\tau) \, x(t) \, dt$$

where $N(\omega)$ is the power spectra (2.12)

$R = 2T - |\tau|$ The total range of integration of the second integral

We are interested in the statistical average of the power spectrum, and must average the product $x(t+\tau) \cdot x(t)$ in the second integral of equation (2.12). But this is the autocorrelation function, $\phi(\tau)$. Substituting and integrating the second integral yields:

$$\overline{N(\omega)} = \int_{-T}^{+T} e^{-j\omega\tau} \ \phi(\tau) \ (1 - \frac{|\tau|}{2T}) \, d\tau \tag{2.13}$$

Letting $T \rightarrow \infty$ equation (2.13) becomes:

$$\overline{N(\omega)} = \int_{-\infty}^{+\infty} \phi(\tau) \, e^{-j\omega\tau} \, d\tau \tag{2.14}$$

The inverse of equation (2.14) is:

$$\phi(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \overline{N(\omega)} \, e^{j\omega\tau} \, d\omega \tag{2.15}$$

If τ is set equal to zero in equation (2.15) we have:

$$\phi(0) = \sigma^2 + \mu^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \overline{N(\omega)} \, d\omega \tag{2.16}$$

From equation (2.16) it is seen that the noise power (or mean square value)

is equal to the sum of the power components at all frequencies.

In the development of the design of the optimal filter, we shall use the term covariance. The covariance function is identical with the autocorrelation function when the means are zero; the latter being true with the Gaussian distribution. The term Noise-to-Signal Power ratio shall also be used. As shown in the preceeding paragraph, this ratio will be the ratio of autocorrelation or covariance functions.

Extension of Statistical Concepts to Vector Quantities.

If we consider a system with multiple inputs excited by Gaussian noise, the covariance function is expressed in matrix notation. We shall concern ourselves with the covariance function when $\tau = 0$. Thus:

$$\phi_{\underline{W}\,\underline{W}^T}(\tau = 0) = E(\underline{W}\,\underline{W}^T) = \begin{bmatrix} \sigma_{w_1}^2 & \sigma_{w_1}\cdot\sigma_{w_2} & \sigma_{w_1}\cdot\sigma_{w_3} & \cdots \\ \sigma_{w_2}\cdot\sigma_{w_1} & \sigma_{w_2}^2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ & & & \sigma_{w_n}^2 \end{bmatrix}$$

where the notation $\phi_{\underline{W}\,\underline{W}^T}$ indicates an autocorrelation on the random input $\underline{W}$.

The terms along the main diagonal of the covariance matrix of excitation noise are the variances (or autocorrelation functions) of the individual noise sources. All other terms of the symmetric matrix are the cross-correlation terms. If the prerequisites of an electronic/electrical control system so dictate, it is attempted to keep various noise sources and deterministic inputs independent of one another in order that the cross-correlation will be zero.

In the development of the optimum filter in a succeeding section we will consider only a single input, and therefore the covariance of excitation will be a scalar.

3. Mathematical description of the plant.

It is assumed that the plant is a linear, time-invariant dynamic process in which the observable states are measured at discrete instances in time. The behavior of the plant is represented by the set of first order differential equations:[1]

$$\dot{\underline{X}} = F\underline{X} + D\underline{u} \qquad (3.1)$$

$$\underline{Y} = B\underline{X} + E\underline{u}$$

where $\underline{X}$ is a real n-vector (n being the system order), the states of the plant: $\underline{u}$ is the control function; F is a real constant n x n matrix; B is a real constant m x n matrix ($m \leq n$); D and E are real constant n-vectors. In this paper, we will deal with situations where we have only one input, thus u becomes a scalar.

It is well known that the solution to equation (3.1) may be written in the form:

$$\underline{X}(t) = \Phi(t - t_o)\, \underline{X}(t_o^+) + \int_{t_o}^{t} \Phi(t - \tau)\, D\, u(\tau)\, d\tau \qquad (3.2)$$

where $\Phi(t - t_o)$ is known as the transition matrix and the integral is one form of the convolution integral.

[1]Equations (3.1) may be encountered in the literature as follows: $\dot{\underline{X}} = A\underline{X} + B\underline{u}$ and $\underline{Y} = C\underline{X} + D\underline{u}$

If the control u is held constant between sampling intervals, equation (3.2) becomes:

$$\underline{X}\left[(k+1)T\right] = \Phi(T)\underline{X}(kT) + \underline{\Delta}(T)\, u(kT) \qquad (3.3)$$

where k indicates the sample number, T the time between samples, and thus kT indicates the sample instant. For notational simplicity T is assumed as "understood" and equation (3.3) is written:

$$\underline{X}(k+1) = \Phi \underline{X}(k) + \underline{\Delta}\, u(k) \qquad (3.4)$$

In block diagram form, equation (3.4) can be represented by:

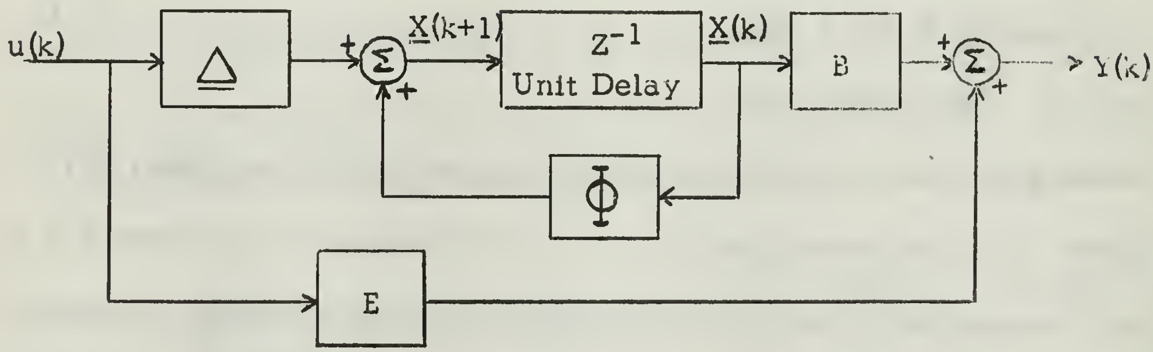

Figure 3-1. Block Diagram of a linear discrete system.

4. Dynamic Programming.

The fundamental principle of dynamic programming is the "Principle of Optimality". This principle states that any portion of an optimal trajectory is also an optimal trajectory. For example if we find the optimum control u_2^o (where the superscript ^{o} denotes an optimal control and the subscript $_2$ denotes the control sequence number), to move from a point X_2 on the optimum trajectory to another point X_3; then the optimum trajectory from X_2 to X_3 will be part of the optimum trajectory from X_1 to X_3.

This principle of optimality is based upon the concept of "embedding"

which says that to solve a specific optimal decision process (with fixed initial time and fixed initial state), the problem is embedded within a group of similar problems (with variable initial states and variable initial time) which are easier to solve. Thus, if we solve the problem for all initial states at $t = t_o$, we will then know the optimal control strategy for the given particular initial states.

If we have a multistage decision process, we can then reduce the multistage decision process to a sequence of single stage decisions. The optimum design problem involves the determination of the sequence of controls $u_o, u_1, u_2, \ldots u_{N-1}$ such that the value of some cost or performance index is minimized, subject to the constraints and dynamics of the system.

Time-Invariant, Linear, Control Processes

It was shown in section 3 that the solution to equation (3.1) is for a sampled system with u(k) constant, the discrete difference equation:

$$\underline{X}(k+1) = \Phi \underline{X}(k) + \underline{\Delta}\, u(k) \tag{4.1}$$

Let us consider the determination of the optimum control law which minimizes the quadratic performance index of the form:

$$J = \sum_{k=1}^{N} \underline{X}_k^T Q \underline{X}_k + R u_{k-1}^2 \tag{4.2}$$

where Q is an n x n positive-definite, symmetric, constant matrix and R is a positive scalar. The versatility of the quadratic performance index given in equation (4.2) is readily apparent. By suitable assignment of values to the elements of the Q matrix, any one of the state variables can

be made more important in specifying the control performance than any other variable. Likewise, suitable choice of R will impose the desired energy constraints on the control signal.

Ogden [5] showed that the dynamic programming approach can be used to formulate a set of recursive relationships which in turn can be programmed on a digital computer to yield the sequence of optimal controls which minimize equation (4.2) subject to the chosen values of Q and R. If we consider an N-stage process (i.e. there are N decisions to be made, $u_o, u_1, \ldots, u_{N-1}$) we can define, by starting at the terminal state first:

$$J_1 = \underline{X}_N^T \, Q \, \underline{X}_N + R\, u_{N-1}^2 \tag{4.3}$$

and we wish to minimize J_1 over u_{N-1}. We can further define using equation (4.1)

$$J_1 = (\underline{\Phi}\, \underline{X}_{N-1} + \underline{\Delta}\, u_{N-1})^T Q\, (\underline{\Phi}\, \underline{X}_{N-1} + \underline{\Delta}\, u_{N-1}) + R\, u_{N-1}^2 \tag{4.4}$$

which becomes:

$$J_1 = \underline{X}_{N-1}^T \underline{\Phi}^T Q \underline{\Phi}\, \underline{X}_{N-1} + u_{N-1}^T \underline{\Delta}^T Q \underline{\Phi}\, \underline{X}_{N-1} + \underline{X}_{N-1}^T \underline{\Phi}^T Q \cdot \underline{\Delta}\, u_{N-1} + u_{N-1}^T \underline{\Delta}^T Q\, \underline{\Delta}\, u_{N-1} + R u_{N-1}^2 \tag{4.5}$$

Since u is a scalar, it is noted that the second and third terms of equation (4.5) are equal. We can thus rewrite equation (4.5) as:

$$J_1 = \underline{X}_{N-1}^T \underline{\Phi}^T Q \underline{\Phi}\, \underline{X}_{N-1} + 2u_{N-1} \underline{\Delta}\, Q\, \underline{\Phi}\, \underline{X}_{N-1} + (\underline{\Delta}^T Q\, \underline{\Delta} + R) \cdot u_{N-1}^2 \tag{4.6}$$

To solve for the optimal control over the last stage, u_{N-1}^o, we must take the partial derivative of J_1 with respect to u_{N-1}, set the result equal to

zero and solve for u_{N-1} which is then u^o_{N-1}.

$$\frac{\partial J_1}{\partial u_{N-1}} = 0 + 2\,\underline{\Delta}\,Q\,\Phi\,\underline{X}_{N-1} + 2(\underline{\Delta}^T Q\,\underline{\Delta} + R)\,u_{N-1} = 0$$

Solving for the optimal control over the last stage yields:

$$u^o_{N-1} = \frac{-\underline{\Delta}^T P_o \Phi}{\underline{\Delta}^T P_o\,\underline{\Delta} + R}\,\underline{X}_{N-1} \triangleq \underline{a}_1^T\,\underline{X}_{N-1} \qquad (4.7)$$

where $P_o = Q$

The performance index now becomes (upon substitution of equation (4.7) into equation (4.6))

$$J_1^o(\underline{X}_{N-1}) = \left[\Phi\underline{X}_{N-1} + \underline{\Delta}\,\underline{a}_1^T\,\underline{X}_{N-1}\right]^T P_o \left[\Phi\underline{X}_{N-1} + \underline{\Delta}\,\underline{a}_1^T\,\underline{X}_{N-1}\right] + \underline{X}_{N-1}^T R\,\underline{a}_1\,\underline{a}_1^T\,\underline{X}_{N-1}$$

Or

$$J_1^o(\underline{X}_{N-1}) = \underline{X}_{N-1}^T\left[\Phi + \underline{\Delta}\,\underline{a}_1^T\right]^T P_o \left[\Phi + \underline{\Delta}\,\underline{a}_1^T\right]\underline{X}_{N-1} + \underline{X}_{N-1}^T R\,\underline{a}_1\,\underline{a}_1^T\,\underline{X}_{N-1} \qquad (4.8)$$

Letting $\Phi + \underline{\Delta}\,\underline{a}_1^T = \Psi_1$, equation (4.8) becomes:

$$J_1^o = \underline{X}_{N-1}^T\left[\Psi_1^T P_o \Psi_1 + R\,\underline{a}_1\,\underline{a}_1^T\right]\underline{X}_{N-1} \qquad (4.9)$$

We can now consider the optimal trajectory during the (N-2)nd stage and determine the optimal control u^o_{N-2}, for that stage. We can denote the optimal (or minimum) cost performance index as J_2^o. It is noted that the subscripts on the J's correspond to the number of the decision being made, and thus the subscripts increase with decreasing time. Let the optimal cost performance over the next-to-last stage be given by:

$$J_2^o = \underset{u_{N-2}}{\text{Minimum}}\left[\underline{X}_{N-1}^T Q\,\underline{X}_{N-1} + R\,u^2_{N-2} + J_1^o\right] \qquad (4.10)$$

Substituting equation (4.9) into equation (4.10) and collecting common

terms yields:

$$J_2^o = \underset{u_{N-2}}{\mathrm{MIN}} \Big[\underline{X}_{N-1}^T (\psi_1^T P_o \psi_1 + Q + R \underline{a}_1 \underline{a}_1^T) \underline{X}_{N-1}$$

$$+ R u_{N-2}^2 \Big] \tag{4.11}$$

Letting $(\psi_1^T P_o \psi_1 + Q + R \underline{a}_1 \underline{a}_1^T) = P_1$ equation (4.11) simplifies to:

$$J_2^o = \underset{u_{N-2}}{\mathrm{MIN}} \Big[\underline{X}_{N-1}^T P_1 \underline{X}_{N-1} + R u_{N-2}^2 \Big] \tag{4.12}$$

Using the recursive relationship given in equation (4.1) we can substitute the value for $\underline{X}_{N-1}$ in equation (4.12) yielding:

$$J_2^o = \underset{u_{N-2}}{\mathrm{MIN}} \Big[(\Phi \underline{X}_{N-2} + \underline{\Delta} u_{N-2})^T P_1 (\Phi \underline{X}_{N-2} + \underline{\Delta} u_{N-2})$$

$$+ R u_{N-2}^2 \Big] \tag{4.13}$$

Using the same arguments as those that preceeded equation (4.6) we can see that equation (4.13) becomes:

$$J_2^o = \underset{u_{N-2}}{\mathrm{MIN}} \Big[\underline{X}_{N-2}^T \Phi^T P_1 \Phi \underline{X}_{N-2} + 2u_{N-2} \underline{\Delta}^T P_1 \Phi \underline{X}_{N-2}$$

$$+ (\underline{\Delta}^T P_1 \underline{\Delta} + R) u_{N-2}^2 \Big] \tag{4.14}$$

Taking the partial derivative of the bracketed term with respect to u_{N-2}, setting the result equal to zero and solving for u_{N-2}^o yields:

$$u_{N-2}^o = \frac{-\underline{\Delta}^T P_1 \Phi}{\underline{\Delta}^T P_1 \underline{\Delta} + R} \underline{X}_{N-2} \triangleq \underline{a}_2^T \underline{X}_{N-2} \tag{4.15}$$

Substituting equation (4.15) into equation (4.13) yields:

$$J_2^o = (\Phi \underline{X}_{N-2} + \underline{\Delta} \underline{a}_2^T \underline{X}_{N-2})^T P_1 (\Phi \underline{X}_{N-2} + \underline{\Delta} \underline{a}_2^T \underline{X}_{N-2})$$

$$+ R \underline{X}_{N-2}^T \underline{a}_2 \underline{a}_2^T \underline{X}_{N-2} = J_2^o (\underline{X}_{N-2}) \tag{4.16}$$

Upon rearranging and collecting terms equation (4.16) becomes:

$$J_2^o = \underline{X}_{N-2}^T \left[\Phi + \underline{\Delta}\, \underline{a}_2^T \right]^T P_1 \left[\Phi + \underline{\Delta}\, \underline{a}_2^T \right] \underline{X}_{N-2}$$

$$+ R\, \underline{X}_{N-2}^T\, \underline{a}_2\, \underline{a}_2^T\, \underline{X}_{N-1} \qquad (4.17)$$

Let $\Psi_2 = \Phi + \underline{\Delta}\, \underline{a}_2^T$, then equation (4.17) simplifies to

$$J_2^o = \underline{X}_{N-2}^T\, \Psi_2^T\, P_1\, \Psi_2\, \underline{X}_{N-2} + R\, \underline{X}_{N-2}^T\, \underline{a}_2\, \underline{a}_2^T\, \underline{X}_{N-2} \qquad (4.18)$$

We have thus developed the optimum trajectory over the last two stages. Should we continue back one more stage, it should be apparent that $P_2 = \Psi_2^T P_1 \Psi_2 + Q + R\, \underline{a}_2\, \underline{a}_2^T$. The general recursion formulas are then

$$\underline{a}_i^T = \frac{-\underline{\Delta}^T P_{i-1}\, \Phi}{\underline{\Delta}^T P_{i-1}\, \underline{\Delta} + R} \qquad (4.19)$$

$$\Psi_i = \Phi + \underline{\Delta}\, \underline{a}_i^T \qquad (4.20)$$

$$P_i = \Psi_i^T P_{i-1} \Psi_i + Q + R\, \underline{a}_i\, \underline{a}_i^T \qquad (4.21)$$

$$P_o = Q,\ \underline{a}_o^T = 0,\ \Psi_o = 0 \qquad (4.22)$$

As noted above, suitable choice of the elements of the Q matrix will determine relative weights of the values of the state variables in the cost function. Likewise, the value of R will determine the emphasis placed upon the energy or input term in calculating the cost performance index. The choice of appropriate values of Q and R is a difficult task since it relies upon engineering judgement.

Ogden [5] presented three cases, selecting different values of Q and R in each case in order to fulfill his stipulated objectives. These cases are repeated below.

CASE I Minimize the terminal states or CEP for the time optimal controller.

Let $Q = I$, the identity matrix $P_o = I$

$R = O$

then $J_N = \text{minimum } \underline{X}_N^T \underline{X}_N$

<u>CASE II</u> Minimize the terminal states with energy constraints over each stage.

Let $Q = I$, the identity matrix therefore $P_o = I$

$R = 1$

then $J_N = \text{minimum } \underline{X}_N^T \underline{X}_N + \sum_{k=1}^{k=N} u_{k-1}^2$

<u>CASE III</u> Minimize the sum of the squares of the states with an energy limitation.

Let $Q = I$, the identity matrix therefore $P_o = I$

$R = 1$

then $J_N = \text{minimum } \sum_{k=1}^{k=N} (\underline{X}_k^T \underline{X}_k + u_{k-1}^2)$

Demetry [2] developed a scheme for selecting Q and R based upon the selection of a desired system characteristic equation for a continuous system. If the polynomial takes the form:

$$s^n + c_n s^{n-1} + c_{n-1} s^{n-2} + \dots + c_1 = 0$$

then Demetry [2] defined Q and R as follows:

$$Q = \begin{bmatrix} \alpha_1 & & & 0 \\ & \alpha_2 & & \\ & & \ddots & \\ 0 & & & \alpha_n \end{bmatrix} \quad \text{and } R = \frac{1}{c_1^2}$$

where $\alpha_j = \frac{1}{c_1^2} \left[c_j^2 + 2 \sum_{m=1}^{n-1} (-1)^m c_{j-m} \cdot c_{j+m} \right]$

and $c_o = 0$

$c_{n+1} = 1$

$c_{(n+1+\ell)} = 0$ for $\ell > 0$

The above values of Q and R are then utilized in Case III above in order to determine the proper feedback matrix to obtain a specific pole-zero configuration. In the illustrative examples we will use this method to obtain a sampled data response which is similar to a continuous response obtained utilizing classical methods of compensation.

Program OPTCON is listed and fully explained in Appendix II as a means for obtaining the optimum feedback matrix $\underline{a}^T$ utilizing the recursive relationships listed in equations (4.19), (4.20), (4.21) and (4.22).

5. The Dynamic Process contaminated with Random Excitation and Measurement Noise.

It is now assumed that the plant is excited by random noise (Gaussian, with zero mean and variance σ^2), W(t). As we have assumed only one excitation, W(t) is a scalar. The measurable output of the system is denoted by the p-vector Y(t), thus:

$$\underline{Y}(t) = \underline{H}(t)\underline{X}(t) \tag{5.1}$$

where H(t) is a p x n matrix which is used as a mechanism for imposing constraints on observing the states of the system. It is further assumed that the measurable output, Y(t), is contaminated by random noise V(t) (also a scalar when only one state or linear combination of states is mea-

sured), yielding a noisy observable Z(t), such that:

$$\underline{Z}(t) = \underline{Y}(t) + \underline{V}(t) \tag{5.2}$$

or:

$$\underline{Z}(t) = H(t)\underline{X}(t) + \underline{V}(t) \tag{5.3}$$

In block diagram, the discretized system now becomes:

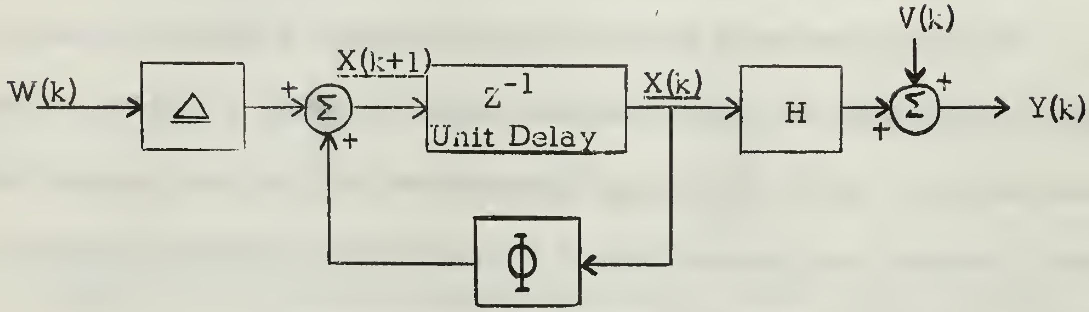

Figure 5-1. Block diagram of system as contaminated by excitation and measurement noise.

6. The Optimum Filter.

It is observed from equation (5.1) that all the state variables are not always explicitly available as measurable quantities. The measurable quantity may be a single state or a linear combination of several states. Thus a device must be designed that will yield the best estimates of the state variables in order to use the feedback values calculated in Program OPTCON. This device is the optimal filter.

It will be shown that the optimal filter is a feedback system, and is obtained by constructing a model (mirror image) of the plant, and obtaining all of the state variables for the plant as an output.

In general it may be said that the best estimate of the states (based on the present observation of the noisy measurable quantities) is equal to

the predicted value of the states (based on the previous observation) plus a correction factor due to the difference between the present noisy observations and the predicted noisy observations. Then we can write:

$$\underline{X}^*(t) = \hat{\underline{X}}(t) + G(t)\left[\underline{Z}(t) - \hat{\underline{Z}}(t)\right] \tag{6.1}$$

where $\underline{X}^*(t)$ is the best estimate of $X(t)$ based on the present observation $\underline{Z}(t)$

$\hat{\underline{X}}(t)$ is the best estimate of $\underline{X}(t)$ based on the previous observation $\underline{Z}(t-1)$

$\hat{\underline{Z}}(t)$ is the best estimate of $\underline{Z}(t)$ based on the previous observation $\underline{Z}(t-1)$

$G(t)$ is the correction factor called the filter gain.

The filter problem then becomes one of the computation of the optimal time-varying gain $G(t)$. This in turn becomes the problem of solving equation (6.1). With a knowledge of the statistical properties of the Gaussian random process, it is possible to determine expressions for the quantities $\hat{\underline{X}}(t)$, $\hat{\underline{Z}}(t)$, and $G(t)$ on the right side of equation (6.1).

Consider $\hat{\underline{X}}(t)$, the best estimate of $\underline{X}(t)$ given $\underline{Z}(t-1)$.

$$\hat{\underline{X}}(t) = E\left[\underline{X}(t)/\underline{Z}(t-1)\right]$$

$$= \Phi(t, t-1)\underline{X}^*(t-1) + \underline{\Delta}(t, t-1)\, E\left[W(t-1)\right] \tag{6.2}$$

But since the mean of the Gaussian excitation noise is zero, we can say:

$$\hat{\underline{X}}(t) = \Phi(t, t-1)\underline{X}^*(t-1). \tag{6.3}$$

Consider $\hat{\underline{Z}}(t)$, the best estimate of $\underline{Z}(t)$ given $\underline{Z}(t-1)$.

$$\hat{\underline{Z}}(t) = E\left[\underline{Z}(t)/\underline{Z}(t-1)\right]$$

However we were given that $\underline{Z}(t) = H(t)\underline{X}(t) + \underline{V}(t)$. Thus:

$$\hat{\underline{Z}}(t) = H(t)\, E\left[\underline{X}(t)/\underline{Z}(t-1)\right] + E\left[\underline{V}(t)\right]$$
$$= H(t)\, \hat{\underline{X}}(t) \qquad (6.4)$$

We now have, except for the gain matrix G(t), expressions for the solution of equation (6.1), and the matrix block diagram of the filter becomes, in the sampled-data case:

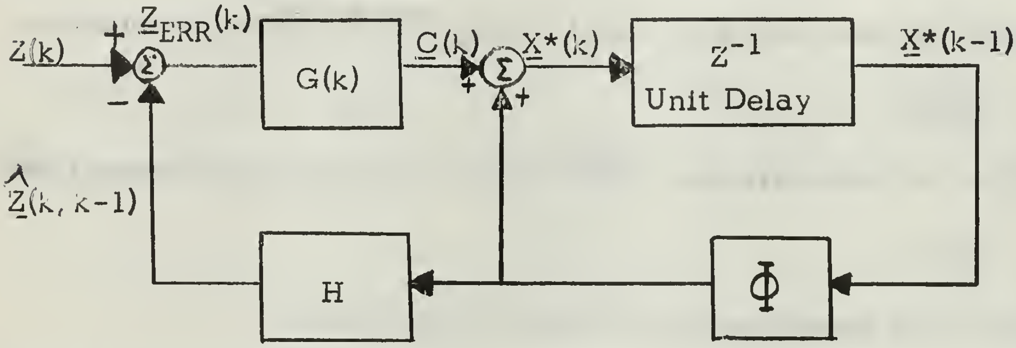

Figure 6-1. Matrix Block Diagram of the filter.

For the complete solution of equation (6.1), an expression for the optimal gain matrix, G(t) must be found. Let us define

$Q = \underline{\Delta} \cdot E\left[\underline{W}(t)\underline{W}^T(t)\right] \cdot \underline{\Delta}^T = \underline{\Delta}\, \sigma_w^2 \underline{\Delta}^T$ the covariance matrix of excitation noise

$R = E\left[\underline{V}(t)\, \underline{V}^T(t)\right]$ the covariance matrix of measurement noise

$P = E\left\{\left[\underline{X}(t) - \hat{\underline{X}}(t)\right]\left[\underline{X}(t) - \hat{\underline{X}}(t)\right]^T\right\}$ the convariance matrix of error in the filter

where the superscript T indicates the transpose of the matrix. It should also be noted that with the introduction of the above notation, we have a duplication of the notation utilizing Q and R both in the controller problem as part of the cost function notation and again as defined above in the filter problem. This duplication is unfortunate, but in keeping with the

notations used in the literature.

It has been shown in the literature [4] that the criterion for finding the optimal gain matrix is to minimize the trace of the convariance matrix of error in estimate. That is, to minimize the loss function:

$$\begin{aligned} L &= \text{Trace } P(t) \\ &= E\left\{\left[\underline{X}(t) - X^*(t)\right]^T \left[X(t) - X^*(t)\right]\right\} \end{aligned} \qquad (6.5)$$

Jardine [3] has shown that after minimizing the loss function given by equation (6.5), the following recursive relationships are valid:

$$G(k) = P(k)\,H^T\left[HP(k)H^T + R\right]^{-1} \qquad (6.6)$$

$$P(k+1) = \Phi\left[P(k) - G(k)HP(k)\right]\Phi^T + Q \qquad (6.7)$$

We have found expressions for all of the quantities on the right side of equation (6.1) and thus have stated the filter problem completely.

Appendix I contains a digital computer program (a modification of Jardine's [3]) which given the F and D matrices, will calculate the optimal gain matrix, G(t). It has been shown that the optimal gain matrix settles out to a stable value after sufficient iteration.

Further description of the use of the program may be found in Appendix I.

7. Closing the Loop.

Section 4 showed how to provide an optimal feedback control around a system whose desired response dictated the values of the terms within a quadratic cost performance index; the latter to be minimized by the feedback controller u. It was shown that this optimal feedback loop is a linear

function of the states, i.e. $u = \underline{a}^T\underline{X}$. In section 6 it was shown that an optimum filter could be designed to give the best estimate of all the states based on observation of only one noisy state and on the knowledge of the system dynamics. Thus we can take the output from the filter $\underline{X}^*$ and premultiply it by $\underline{a}^T$ to complete the feedback loop.

We now consider the problem of imposing a deterministic input upon the system. Let the deterministic input be represented by the vector $\underline{DI}(k)$, where the elements are the input itself and its successive derivatives. Demetry [1] showed that $\underline{DI}(k)$ must be introduced so as to affect the filter and plant operation in identically the same manner. Figure 7-1 shows a block diagram of the manner in which $\underline{DI}(k)$ is introduced into the complete plant-filter-controller system. An inspection of figure 7-1 reveals the following recursive formulae:

$$\begin{aligned}
\underline{Y}(k) &= H\underline{X}(k) \\
\underline{Z}(k) &= \underline{Y}(k) + \underline{V}(k) \qquad\qquad (7.1) \\
\underline{X}^*(k) &= (I-GH)\Phi\,\underline{X}^*(k-1) + (I-GH)\,\underline{\Delta}\,\underline{a}^T\left[\underline{X}^*(k-1) - \underline{DI}(k-1)\right] \\
&\quad + G\underline{Z}(k) \\
\underline{X}(k+1) &= \Phi\,\underline{X}(k) + \underline{\Delta}W(k) + \underline{\Delta}\,\underline{a}^T\left[\underline{X}^*(k) - \underline{DI}(k)\right]
\end{aligned}$$

Program CLOSE is designed to solve equations (7.1). (See Appendix III)

A further discussion of the make-up of the vector $\underline{DI}(k)$ is necessary, however it will be deferred to section 8 and discussed by example utilizing the illustrative examples chosen for this paper.

We now have the complete system of plant, filter, and controller

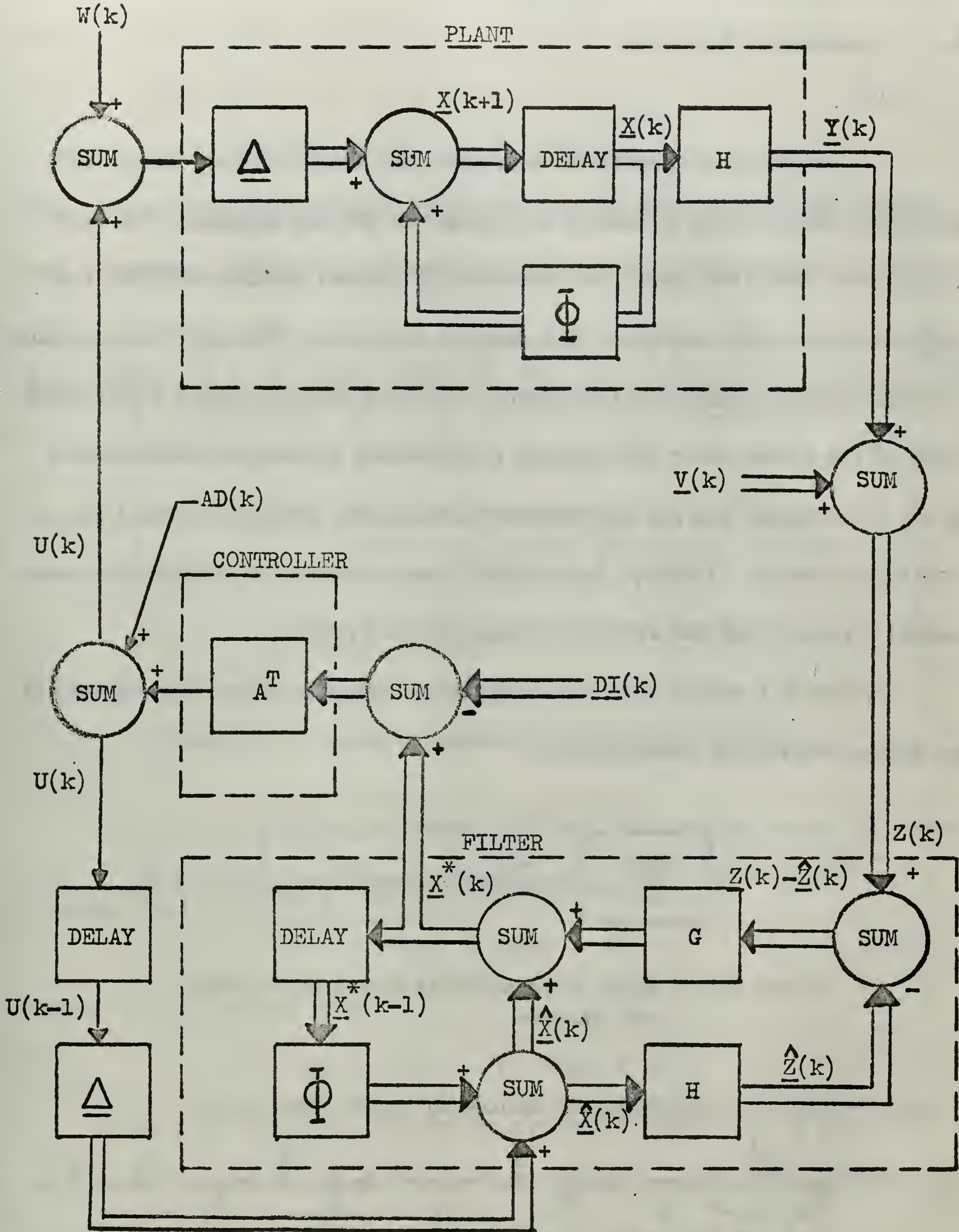

Fig 7-1 Filter Controller Schematic

with deterministic inputs. Illustrative examples of the use of these concepts are presented in the following section.

8. Illustrative Examples.

I.

As the first illustrative example, we shall consider an aircraft autopilot device used to control an aircraft in the roll attitude. To this system we shall first apply the concepts of optimal control (OPTCON) to stabilize the system and give us a desired response. We will then consider the design of an optimal filter for the system in order to obtain a best estimate of the states when the aircraft is perturbed by random disturbances (i.e. wind gusts) and the measurement device has inherent errors (i.e. a noisy gyroscope). Finally, the control loop is closed to provide for deterministic inputs and the over-all control of the system.

Figure 8-1 shows the open-loop block diagram of the roll channel of an aircraft-autopilot combination.

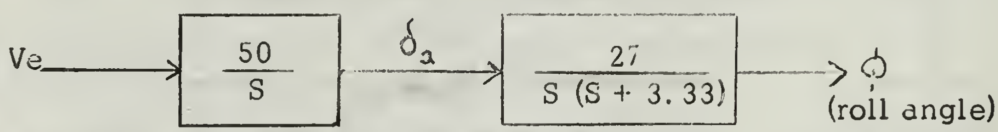

Figure 8-1. Block Diagram of an autopilot/aircraft roll channel.

Figure 8-1 can be reduced to the following signal flow graph.

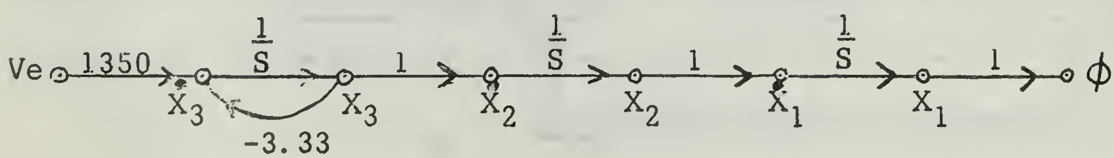

Figure 8-2. Signal Flow Graph of Autopilot System.

Letting V_e equal u, and noting that X_1 is equal to ϕ, the system can be represented by the vector-matrix differential equation:

$$\dot{\underline{X}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -3.33 \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 0 \\ 1350 \end{bmatrix} u \qquad (8.1)$$

This system has been compensated [6] to obtain stability and the desired root locations. Dominant mode design gives the system shown in figure 8.3.

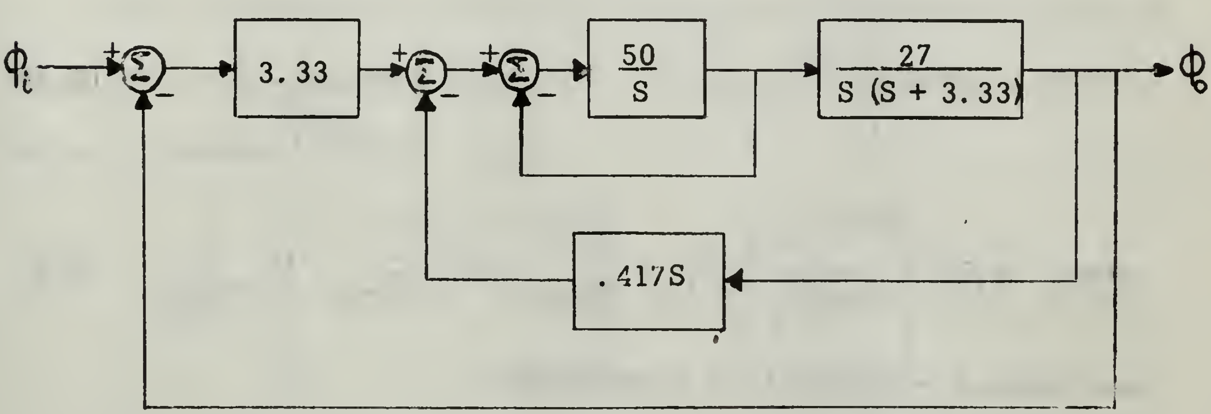

Figure 8-3. Block diagram of Compensated System.

Using block diagram reduction methods, the system shown in Figure 8-3 yields the following characteristic equation:

$$s^3 + 53.3s^2 + 729s + 4495 = 0 \qquad (8.2)$$

Equation (8.2) factors into:

$$(s + 37.6)(s + 8.25 + j7.39)(s + 8.25 - j7.39) = 0 \quad (8.3)$$

It is seen that the compensated system of Figure 8-3 is dominated by the pair of complex conjugate roots and is thus essentially a second-order system with $\zeta = .745$, $\omega_n = 11.06$, and a time constant equal to 0.121

seconds. The step response to this continuous system is shown in Figure 8-4.

We now wish to determine the stable feedback gain matrix $\underline{a}^T$, which, when applied to the uncompensated system shown in Figure 8-1, will yield a system with the characteristic equation given in equation (8.2). Referring to section 4, and using Demetry's [1] procedure of choosing Q and R in the cost function yields:

$$Q = \begin{bmatrix} 1 & & \\ & \alpha_1 & \\ & & \alpha_2 \end{bmatrix} \qquad R = \frac{1}{C_1^2} \tag{8.4}$$

$$\text{where} \quad \alpha_j(n) = \frac{1}{C_1^2}\left[C_j^2 + \sum_{m=1}^{n-1} (-1)^m \, C_{j-m} \cdot C_{j+m}\right] \tag{8.5}$$

and where n = order of the system and:

$$\begin{aligned} C_o &= 0 \\ C_{n+1} &= 1 \\ C_{n+1+\ell} &= 0 \text{ for } \ell > 0 \end{aligned} \tag{8.6}$$

The coefficients of the characteristic equation of the compensated system are:

$$\begin{aligned} C_1 &= 4495 \\ C_2 &= 729 \\ C_3 &= 53.3 \\ C_4 &= 1 \end{aligned} \tag{8.7}$$

Substituting these values in equation (8.5) yields the following values for

α:

$$\alpha_1 = 2.53 \times 10^{-3} \tag{8.8}$$

$$\alpha_2 = 6.87 \times 10^{-5}$$

These values in turn give us the following values for Q and R:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2.53\text{x}10^{-3} & 0 \\ 0 & 0 & 6.87\text{x}10^{-5} \end{bmatrix} \qquad R = 4.95 \times 10^{-8} \tag{8.9}$$

With the knowledge of the F and D matrices from equation (8.1), we can calculate Φ, the transition matrix, and Δ, the distribution matrix, for use in program OPTCON. Thus:

$$\Phi = \begin{bmatrix} 1.0 & .050 & .001184 \\ 0 & 1.0 & .046093 \\ 0 & 0 & .847893 \end{bmatrix} \qquad \Delta = \begin{bmatrix} .027002 \\ 1.59839 \\ 62.2253 \end{bmatrix} \tag{8.10}$$

The selection of a sampling interval is not completely arbitrary. It is our purpose here to show that feedback compensation based upon the concepts of program OPTCON (presented in section 4) will yield a response similar to that obtained from the system as compensated in Figure 8.3. Since an increase in the time between samples will normally lead to increased oscillations and decreased stability; a sufficiently small sampling interval is selected in order to closely approximate the continuous system. For this example the sampling interval (denoted by DT in FORTRAN programs) is taken as .05 seconds.

Entering program OPTCON with the values of Q, R, Φ, Δ, and the

sampling interval given, yields the following values for the stable steady state feedback gain matrix:

$$\underline{a}^T = \begin{bmatrix} -1.295 & -.2447 & -.0190 \end{bmatrix} \quad (8.11)$$

We have seen from section 4 that the feedback control is a linear function of the states; that is, $u = \underline{a}^T\underline{X}$. Thus all the states must be available for measurement in order to complete the feedback loops. But, in the example of the autopilot roll control channel, we can measure only the roll angle (which is the state variable X_1). Furthermore, this measurement of roll angle is not exact since there are inaccuracies in the measuring device (a gyro-stabilized platform with appropriate attitude sensors). Also the states are aggravated by a non-deterministic input, wind gusts which are of a random nature. It is apparent that there is need for a device that will give us the best estimates of the state variables based upon the observation of only one noisy state and on knowledge of the system dynamics. Thus we must apply the concepts of the optimal filter to this practical problem.

For the non-deterministic input we have chosen a variance equal to .25, that is, $\sigma_W^2 = E\left[W.W^T\right] = .25$. Using the F matrix and D vector as given in equation (8.1), and the sampling interval of .05 seconds, we have determined the $\underline{\Phi}$ and $\underline{\Delta}$ matrices of the system (see Program FILTALL, Appendix I). The variance and the $\underline{\Delta}$ vector can be combined to yield the covariance matrix of random excitation processes, Q, i.e.:

$$Q = \underline{\Delta}\,\sigma_W^2 \cdot \underline{\Delta}^T \quad (8.12)$$

We have chosen a Noise to Signal Power ratio of .1, where by definition:

$$\frac{\text{Noise Power}}{\text{Signal Power}} = \frac{N}{S} = \frac{r_{11}}{q_{11}} \tag{8.13}$$

where

r_{11} is the first element of the covariance matrix of measurement noise, R, (recall from section 6 that $R = \sigma_V^2 = E[V.V^T]$).

q_{11} is the first element of the covariance matrix of the random excitation process.

It should be noted that for a system where there is only one measurable state, the R matrix reduces to a scalar quantity, that is, $R = r_{11}$. Given the value of Q (and thus q_{11}) and the desired N/S, program FILTALL will compute the necessary value for r_{11} to satisfy equation (8.13).

We can now enter program FILTALL (see Appendix I) with our values of F and D, sampling interval, σ_W^2, and N/S and determine the stable steady state gain matrix G. The optimal steady state G matrix thus computed is:

$$G = \begin{bmatrix} .99384 & 0 & 0 \\ 31.77063 & 0 & 0 \\ 515.40355 & 0 & 0 \end{bmatrix} \tag{8.14}$$

With the value of G (from program FILTALL) and the value of $\underline{a}^T$ (from program OPTCON), we are in a position to enter program CLOSE (see Appendix III) with a deterministic input to determine the system response to that input. In order to compare our response with that of Schmidt [6], the deterministic input was selected as a step of magnitude ten. Figure 8-4

shows the response of our overall system as compared with Schmidt's. It is noted that the responses are quite similar. The rise time of our system is slightly smaller than that of the comparable continuous system. This result might be expected in sampled-data systems.

As indicated in section 7, a further discussion of the vector $\underline{DI}(k)$ is necessary. This discussion is as presented by Demetry [1] and applied to this example.

By inspection of Figure 7-1 it is seen that a vector $\underline{B}$ might be defined as follows:

$$\underline{B}(k) = \underline{X}^*(k) - \underline{DI}(k) \tag{8.15}$$

If it is our desire to drive the vector $\underline{X}$ to the origin of the $\underline{B}$ space, we have in effect a regulator problem. As our plant is a type 2, it will follow only "step" and "ramp" inputs with zero steady-state error. In our example, we required the step response to a step input of 10. Therefore $X_1(\infty) = 10.0$, $X_2(\infty) = X_3(\infty) = 0.0$. This would be accomplished by setting $DI(1) = 10.0$, and $DI(2) = DI(3) = 0.0$ and as is seen by inspection of equation (8.15), the $\underline{B}$ vector in steady state is zero.

If we wanted a "ramp" response, $X_1(\infty) = t$, $X_2(\infty) = 1.0$, and $X_3(\infty) = 0.0$, we would input $DI(1) = t$, $DI(2) = 1.0$ and $DI(3) = 0.0$. Again by inspection these final conditions are compatible with the plant dynamics. The $\underline{B}$ vector would again be driven to the origin of the $\underline{B}$ space.

If we wanted a "parabolic" response, $X_1(\infty) = t^2$, $X_2(\infty) = 2t$, and $X_3(\infty) = 2.0$, we would choose $DI(1) = t^2$, and $DI(3) = 2.0$ hoping again to drive the $\underline{B}$ vector to its origin. However, in the state equation

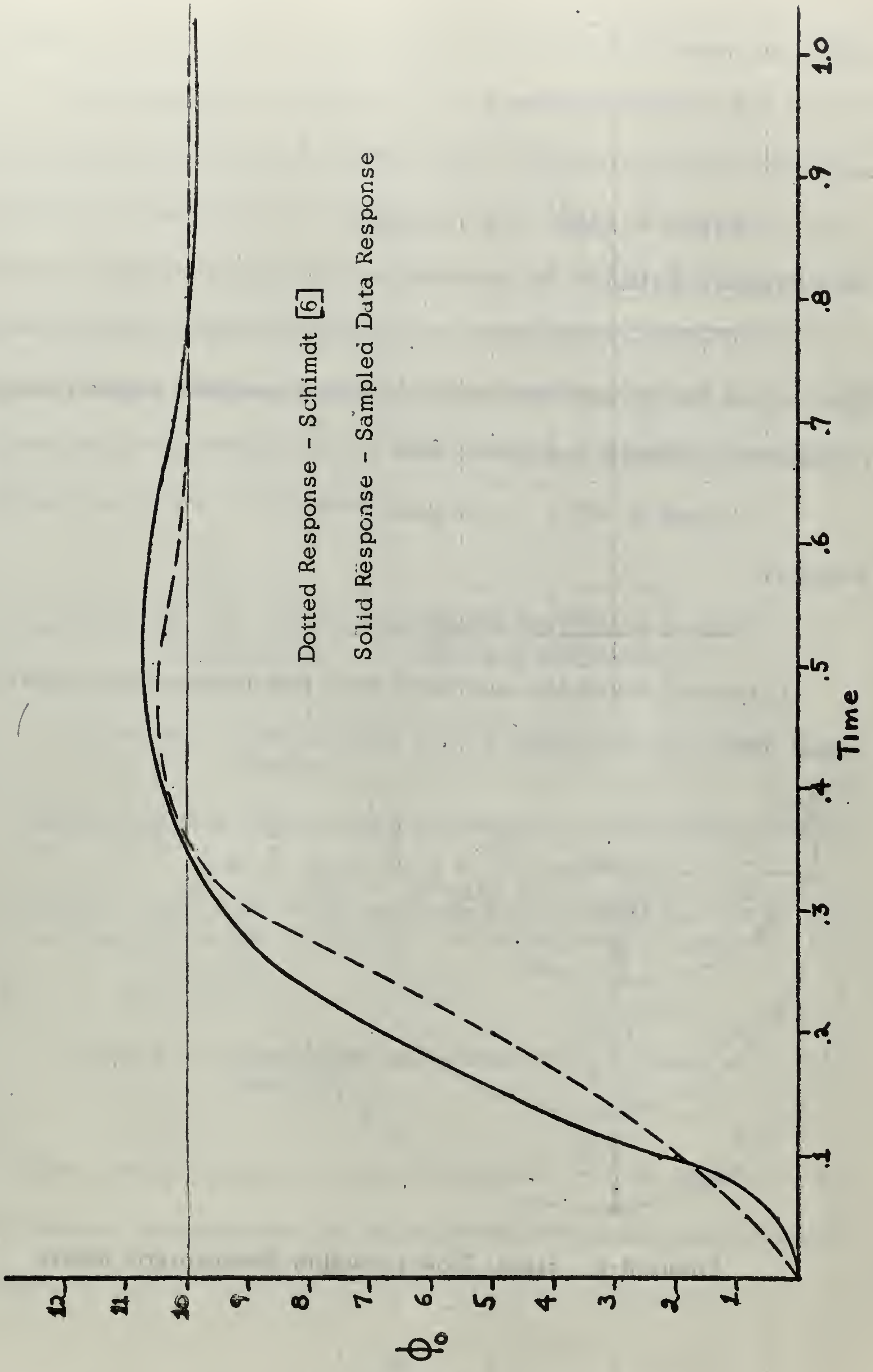

Figure 8-4. Comparison of Step Responses Example I.

for $\dot{X}_3$, we have:

$$\dot{X}_3 = u - 3.33\, X_3$$

and

$$\dot{X}_3(\infty) = \cancel{u(\infty)}^{\,0} - 3.33X_3(\infty)$$

but we require $\dot{X}_3(\infty) = 0$ therefore:

$$0 \neq 0 - 3.33 \cdot (2.0)$$

However we can by-pass this difficulty with a parabolic input by introducing an additional constant input such that:

$$\dot{X}_3(\infty) = AD - 3.33X_3(\infty) = 0$$

therefore:

$$AD = 3.33(2.0) = 6.66$$

Figure 8-5 shows the controlled plant with deterministic inputs in signal flow.

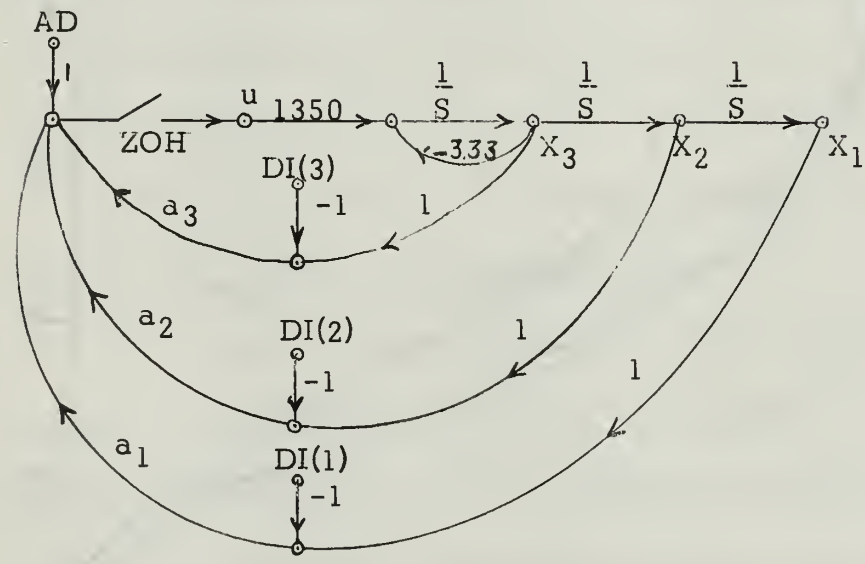

Figure 8-5. Signal Flow including deterministic inputs.

II.

As illustrative example two, we shall consider the same aircraft autopilot device considered in example one. However, we now desire to control the pitch attitude of the aircraft. Again, as in example one, the concepts of optimal control, optimal filtering and study of the closed loop response will be utilized to stabilize and control the pitch attitude and obtain a response which is comparable to a response obtained by Schmidt[6]. The open-loop block diagram of the pitch channel of the aircraft-autopilot combination is shown in Figure 8-6.

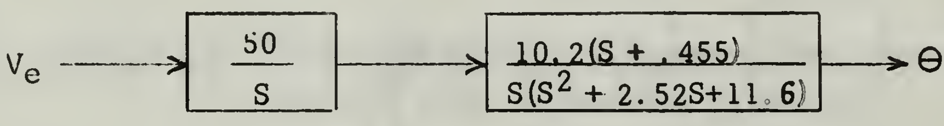

Figure 8-6. Block diagram of an autopilot/aircraft pitch channel.

Figure 8-7 shows the above block diagram in signal flow notation.

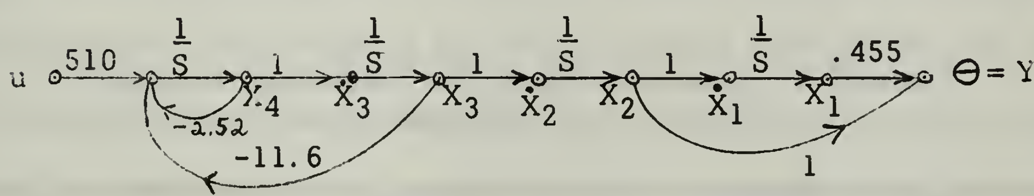

Figure 8-7. Signal Flow representation of the pitch channel

Again, letting V_e equal u, and noting that $\dot{\theta} = Y = .455 X_1 + X_2$, the system can be represented by the following vector-matrix differential equation:

$$\dot{\underline{X}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -11.6 & -2.52 \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 510 \end{bmatrix} u \qquad (8.16)$$

As indicated above, this system has been compensated by Schmidt [6] to obtain stability and desired root locations. Figure 8-8 shows the complete block diagram including parameter constants resulting in a desired response. This response is shown in Figure 8-10 where it is compared with the sampled response.

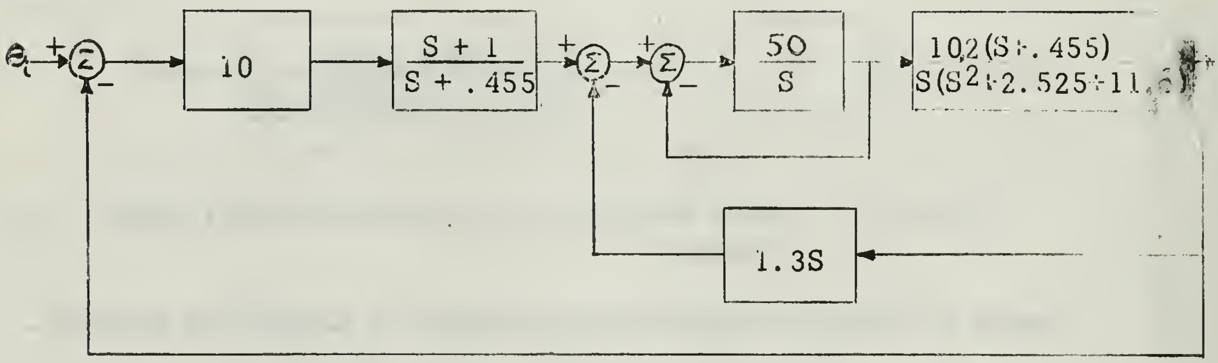

Figure 8-8. Block diagram of the Compensated system

Using block diagram reduction methods, the system shown in Figure 8-8 reduces to the following characteristic equation:

$$s^4 + 52.52\, s^3 + 800\, s^2 + 5981\, s + 5100 = 0 \qquad (8.17)$$

Equation 8.17 factors into:

$$(s + .971)(s + 34.06)(s + 8.7457 \pm j8.817) = 0 \qquad (8.18)$$

It is seen here that the closed loop zero is nearly cancelled in its effect by the closed loop pole (s + .971). This results in a response

which is dominated by the complex roots with little effect on the response being produced by the second pole on the real axis (i.e. dominant second-order design). Figure 8-9 shows the closed loop pole-zero plot of the above situation.

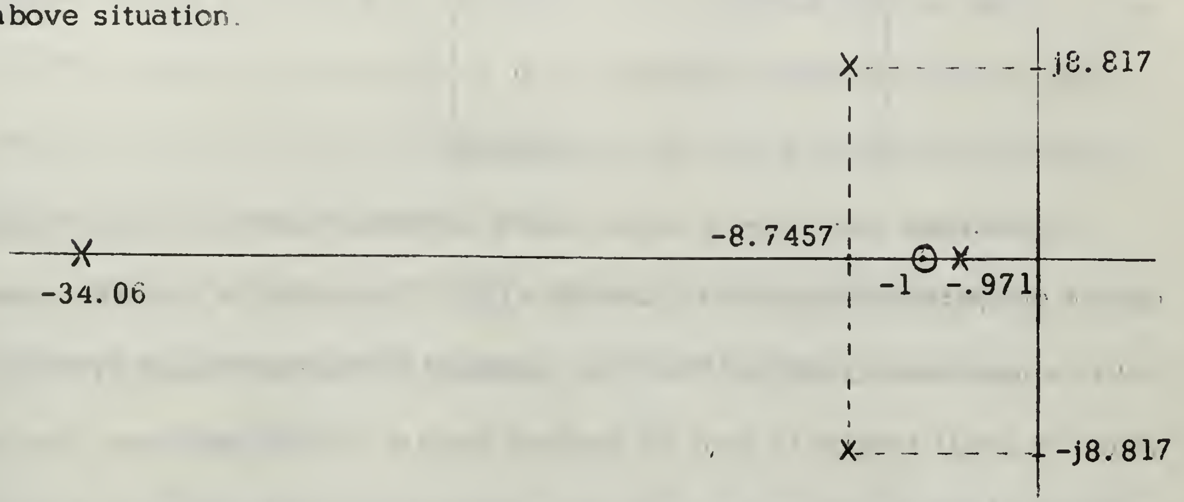

Figure 8-9. Pole-Zero plot of the closed loop transfer function shown in equation 8.18.

Since the system to be controlled by the methods in this paper is that shown in Figure 8-6, it can be seen that a zero is present at -.455 which must be contended with. As we want a system in the final analysis which will yield the same response as was obtained by Schmidt [6], we too, must attempt to obtain a closed loop pole-zero configuration similar to that shown in Figure 8-9, for the continuous system. To accomplish this, we take the three poles which are dominating the response i.e. (s + 34.06) and (s + 8.7457 $\pm$ j8.817) and introduce a fourth pole at (s + .455) which results in the following characteristic equation:

$$s^4 + 52.005\, s^3 + 773.45\, s^2 + 5594\, s + 2390 = 0 \qquad (8\text{-}19)$$

From this equation, Q and R from the cost function may be calculated exactly as illustrated in example one. These values are then utilized as

one input to program OPTCON in the calculation of a transpose, the feedback gain matrix. The resulting Q and R from equation 8.19 are listed below:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4.82 & 0 & 0 \\ 0 & 0 & .0035 & 0 \\ 0 & 0 & 0 & .0002058 \end{bmatrix}, \quad R = 1.75 \times 10^{-7} \qquad (8.20)$$

In example one, it was shown that a sampling interval of .05 seconds gave a comparable response to Schmidt's [6]. This value of sampling interval is approximately one half the time constant of the continuous system, which is small enough to give the desired results. In example two, the time constant of the continuous system is approximately equal to that of example one, therefore the sampling interval in example one will also be satisfactory for example two. Therefore with the F and D matrices listed in equation 8.16, Φ and Δ may be calculated for use in Program OPTCON. These are listed below:

$$\Phi = \begin{bmatrix} 1.0 & .05 & .001247 & .00002016 \\ 0 & 1.0 & .049766 & .0011962 \\ 0 & 0 & .986123 & .0467515 \\ 0 & 0 & -.54231813 & .86831 \end{bmatrix}, \quad \Delta = \begin{bmatrix} .0001294087 \\ .01283806 \\ .6100834 \\ 23.843297 \end{bmatrix} \qquad (8.21)$$

With the above listed values of Φ, Δ, Q, R, and sampling interval, program OPTCON yields the following values for the stable feedback gain matrix:

$$\underline{a}^T = \begin{bmatrix} -1.8320888 & -4.425734 & -.6917466 & -.051754 \end{bmatrix} \tag{8.22}$$

We are now ready to calculate the steady state gain matrix in program FILTALL. As described in example one, we have assumed a noise to signal power ratio of .1, while the statistic of the random excitation noise, σ_W (SIGW), remains as assumed .5. Program FILTALL then calculates the required statistic for the random measurement noise to give the assumed value of noise to signal power ratio.

With the above assumptions, program FILTALL determines the steady state optimal gain matrix G which is listed as follows:

$$G = \begin{bmatrix} .01782 & 0 & 0 & 0 \\ .99189 & 0 & 0 & 0 \\ 33.54936 & 0 & 0 & 0 \\ 582.73566 & 0 & 0 & 0 \end{bmatrix} \tag{8.23}$$

With the value of G (from program FILTALL) and the value of $\underline{a}^T$ (from program OPTCON), we are again in the position to enter program CLOSE with a deterministic step input, and determine the system response. We enter program CLOSE with Φ, Δ, sampling interval, σ_W, $\sigma_V(1)$, G, and $\underline{a}^T$ as indicated above. The output of the program is the time response desired which is shown in Figure 8-10 and compared with the continuous response obtained by Schmidt [6] of the system in Figure 8-8.

Again in example two, we require an additional discussion of the required deterministic inputs because we now have to deal with the zero (s + .455). Here also, the discussion is patterned after Demetry [1] with

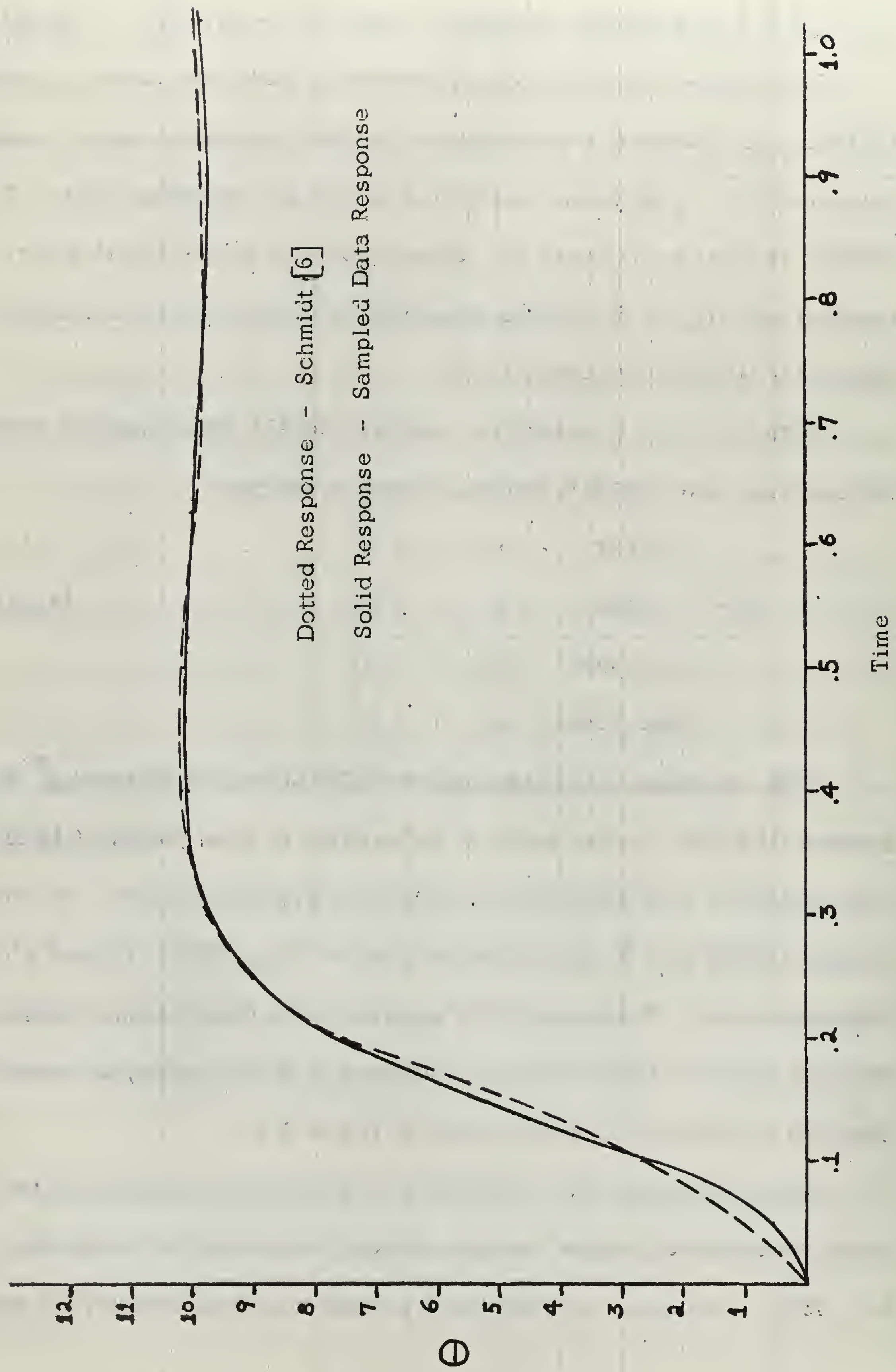

Figure 8-10. Comparison of Step Responses Example II

the numbers from example two substituted.

The output in example two is a linear combination of two states:

$$Y = .455\ X_1 + X_2 \tag{8.24}$$

The required step response is to a step of ten. This being the case, we desire that $Y(\infty) = 10$. For this to be true, $X_1(\infty)$ must equal 1/.0455 and $X_2(\infty)$ must equal zero. Therefore as discussed in example one, the required deterministic inputs would be DI(1) = 1/.0455, DI(2) = DI(3) = DI(4) = 0.0. As can be readily seen the requirements for a ramp or parabolic input would result in a discussion similar to that in example one, therefore it will not be repeated here.

Figure 8-11 shows the complete signal flow diagram including deterministic inputs.

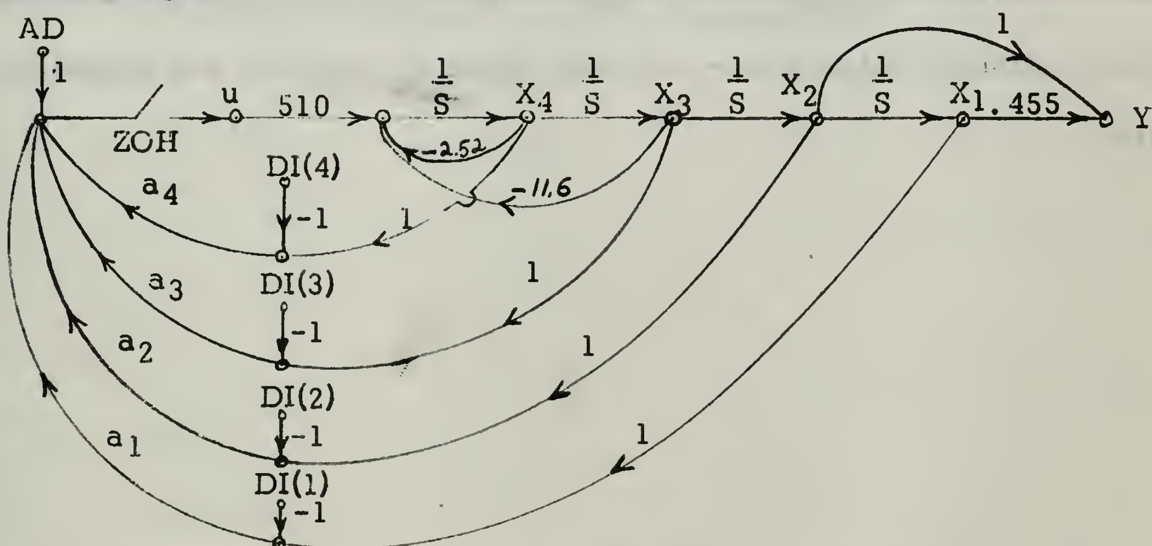

Figure 8-11. Signal Flow Representation including deterministic inputs.

9. Summary and Conclusions.

We have shown the development of an optimal feedback matrix for the

control of any open loop system. Also developed was the optimal filter which is utilized when the system in question is excited by noise and is plagued in the measurement of its output by measurement noise. Both the filter and the optimum controller can be combined in a computer which will be programmed to produce a model of the system process. This model produces the best estimates of the actual system states which are then used to produce the optimum control.

Two illustrative examples demonstrate that the combination controller/ filter is a powerful tool for the control of actual physical systems.

Areas for future work might include the cases of non-stationary plants, non-stationary statistics, and the use of variable filter and controller gains either continuously utilizing an on-line computer, or in stages (an adaptive filter/controller) utilizing pre-computed values for the filter and controller gain.

BIBLIOGRAPHY

1. Demetry, J. S., Strum, R. D., Titus, H. A. A Case Study of the Discrete Optimum Filter-Controller Problem. U. S. Naval Postgraduate School Research Paper Number 57, November 1965.

2. Demetry, J. S. and Titus, H. A. Linear Control System Optimization Using a Model Based Index of Performance. U. S. Naval Postgraduate School Research Paper Number 48, November 1964.

3. Jardine, F. D. Optimal Filter Design for Sampled Data Systems with Illustrative Examples. U. S. Naval Postgraduate School Masters Thesis 1965.

4. Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, March 1960.

5. Ogden, D. J. Optimum Digital Control Systhesis. U. S. Naval Postgraduate School Masters Thesis, 1965.

6. Schmidt, S. F. and Triplett, W. C. Use of Nonlinearities to Compensate for the Effects of a Rate-Limited Servo on the Response of an Automatically Controlled Aircraft. National Advisory Committee for Aeronautics Technical Note 3387, January 1955.

APPENDIX I

Program Filtall is listed below completely. A thorough discussion of the various parts of the program is interspersed throughout the listing. Program statements are prefaced with a P in the margin and are written in FORTRAN. A complete uninterrupted program is also listed at the end of the Appendix.

It is the intent of the authors to provide a complete and detailed FORTRAN program for the solution of the "gain problem" as descirbed in section 6. The variations possible from the listed program will be discussed in detail.

PROGRAM FILTALL

This program as listed is set up for a fourth order system but as will be pointed out as we go, it can be expanded to higher order, or used as is for third and second order systems. The program itself is very general, however as order increases, print out of the desired information becomes more and more complex. Print out examples will not necessarily be substantiating data for the illustrative examples but rather examples which can conveniently be fitted on to the standard $8\frac{1}{2}$ by 11 size paper. Substantiating data for the illustrative examples will be listed with the example.

Data Card Discussion:

A. Data Card 1 N, the order of the system Format I2.

B. We follow Data Card 1 with a set of data cards each set containing 3 cards.

1. Data Card 1(of each set) DT, the sampling interval, Format F10. 0.

2. Data Card 2(of each set) F, F matrix as described in Section .3, Format 8F10. 0.

3. Data Card 3(of each set) D, D matrix as described in Section .3, Format 8F10. 0.

Some comments:

PHI is the system transition matrix, DEL the distribution matrix. G is the optimum gain matrix, H the observable matrix, P the best estimate of the covariance matrix of error, and Q is the excitation noise covariance matrix.

DIMENSION STATEMENTS (Lists all vectors and matrices utilized in main program)

```
P          DIMENSION P (12, 12), Q(12. 12), H(12, 12), R(12, 12), G(12, 12)
P         1, PHIT(12, 12), PHI(12, 12), DEL(12), DELDELT(12, 12), XG11(40),
P         2XG21(40), XG31(40), XG41(40)
```

N, the order of the system as indicated above is now read in.

```
P          READ 33, N
P       33 FORMAT (I2)
```

SIGWSQ is the variance of the random noise excitation, SIGW is the standard deviation of this random source. We have assumed this deviation to be .5 in the outlined program, however this is one parameter which can be studied, as to its effects on the plant, the gain matrix, etc.

```
P          SIGWSQ=.5
```

```
P          SIGW=SQRTF(SIGWSQ)
```

We now come to the first loop of the main program and it encompasses the entire main program. It is a loop designed to vary the value of the sampling interval, DT, and we have provided for 7 values, any number of values may be utilized however, with the proper adjustment of the data cards.

```
P          DO 1001 LK=1,7
```

The sampling interval DT, is now read in.

```
P          READ 2, DT
P        2 FORMAT (F10.0)
```

Subroutine PHIDEL is now called, and within PHIDEL we read in the values of the F matrix and D matrix, and we call the subroutine with N and DT, and the results are PHI the system transition matrix, and DEL the distribution matrix.

```
P          CALL PHIDEL(PHI, DEL, N, DT)
```

Subroutine PHIDEL as described in Ogden [5], accomplished the above results by using the Taylor Series expansion for the exponential function. The calculation of PHI and DEL is good only for non-time varying dynamics because the integration is accomplished within the program for non-time varying dynamics only, ie, F and D must be constant matrices. The entire subroutine is listed below.

```
P          SUBROUTINE PHIDEL(PHI, DEL, N, DT)
P C        VALID ONLY FOR A CONSTANT F MATRIX
P          DIMENSION F(12,12), PHI(12,12), TERM(12,12), WORM(12,12)
P         1, DEL(12), DELM(12,12), TELM(12,12), DELP(12,12), D(12)
```

```
P       1 FORMAT  ((8F10.0))
P         READ1, ((F(IR, IC), IC=1, N), IR=1, N)
P         READ 1 (D(I), I=1, N)
P    1003 PRINT 399, DT, ((F(IR, IC), IC=1, N), IR=1, N)
P     399 FORMAT (1H1, 8X, 3HDT=, 1F5.3////, 7HF(I, J)=/, ((8F8.2)))
P         PRINT 3991 (D(I), I=1, N)
P    3991 FORMAT(////5HD(I)=/(8F8.2))
P         NFINAL=1
P         TM=0.0
P         DO 400 IR=1, N
P         DO 400 IC=1, N
P         TERM(IR, IC)=0.0
P         WORM(IR, IC)=0.0
P         TERM(IR, IR)=1.0
P         TELM(IR, IC)=TERM(IR, IC)*DT
P         DELP(IR, IC)=TELM(IR, IC)
P         DELM(IR, IC)=0.0
P         DEL(IR)=0.0
P     400 PHI(IR, IC)=TERM(IR, IC)
P       4 TM=1.0+TM
P         DO 500  IR=1, N
P         DO 500  IC=1, N
P         DO 500  JN=1, N
```

```
P         DELM(IR, IC)=DELM(IR, IC)-TELM(IR, JN)*F(JN, IC)*DT/(TM+1.0)
P     500 WORM(IR, IC)=TERM(IR, JN)*F(JN, IC)*DT/TM+WORM(IR, IC)
P         DO 401  IR=1, N
P         DO 401  IC=1, N
P         TERM(IR, IC)=WORM(IR, IC)
P         TELM(IR, IC)=DELM(IR, IC)
P         DELP(IR, IC)=DELP(IR, IC)+TELM(IR, IC)
P         PHI(IR, IC)=PHI(IR, IC)+TERM(IR, IC)
P         DELM(IR, IC)=0.0
P     401 WORM(IR, IC)=0.0
P         ABC=0.0
P         DO 2I=1, N
P         DO 2J=1, N
P         AA=TERM(I, J)
P         AB=ABSF(AA)
P         IF(ABC-AB)3, 3, 2
P       3 ABC=AB
P       2 CONTINUE
P         IF(0.000000005-ABC)5, 5, 6
P       5 GO TO 4
P       6 PRINT 502, (PHI(IR, IC), IC=1, N), IR=1, N)
P     502 FORMAT(////9X, 8HPHI(I, J)///(6E20.8))
P         DO 600  I=1, N
```

```
P          DO 600  K=1, N
P          DO 600  J=1, N
P     600  DEL(I)=DEL(I)+PHI(I, J)*DELP(J, K)*D(K)
P          PRINT 503 (DEL(I), I=1, N)
P     503  FORMAT(////9X, 6HDEL(I)//(6E20.8)//)
P          END
```

The first page of print out results from subroutine PHIDEL with DT, F, D PHI and DEL are printed out. This is illustrated as Sample Print Out Page 1.

Next we initialize the measurement noise variance, R(1, 1) prior to starting our next loop. R(1, 1) is a scalar because only one state is observed

```
P          R(1, 1)=0.0
```

The second major loop in the program follows. We might call this our R(1, 1) loop as we increment R(1, 1) from the above initialized value. We have chosen eleven (11) values and these may be varied as desired. We define the noise to signal power ratio (PNS) as R(1, 1)/Q(1, 1). As described in Jardine [3] we will set R(1, 1) equal to multiples of Q(1, 1) in order provide convenient noise to signal power ratios. This second loop also encompasses the entire main program from this point.

```
P          DO 1001 LL=1, 11
```

The next series of statements are utilized to first initialize Q, P, H, and G matrices at zero (0). $\Delta\Delta^T$ (DELDELT) is then calculated.

```
        DT= .050

F(I,J)=
      .00    1.00      .00      .00



D(I)=
      .00    1.00



        PHI(I,J)

        .10000000E+01       .50000000E-01       .00000000E+00       .10000000


        DEL(I)

        .12500000E-02       .50000000E-01
```

Sample print out Page 1
Appendix I

```
P           DO 1001 LL=1, 11
P           DO 699 I=1, N
P           DO 699 J=1, N
P           Q(I, J)=0.
P           P(I, J)=0.
P           H(I, J)=0.
P       699 G(I, J)=0.
P           DO 30 I=1, N
P           DO 30 J=1, N
P        30 DELDELT(I, J)=DEL(I)*DEL(J)
```

The matrix Q is defined as DEL x $E(WW^T)$ x DEL^T. However $E(WW^T)= \sigma_W^2$ a scalar which is one of our assumptions. Therefore $Q = \sigma_W^2 \Delta \Delta^T$ defines Q and we initialize Q next in the program.

```
P           DO 40 I=1, N
P           DO 40 J=1, N
P           Q(I, J)=SIGWSQ*DELDELT(I, J)
```

A new variable XQ is introduced as an aid in incrementing R near the end of the loop, as the original value of Q is lost in a subsequent subroutine.

```
P           XQ=Q(1, 1)
```

One of the more controversial initializations involves the P matrix or the covariance matrix of error. We initialized it by setting it equal to Q, There are many opinions as to the initial value of P, and we have tried

to discuss them in our case studies. This is another point in the program where engineering judgement and prior knowledge of the problem will help in providing a best initial P.

```
P           DO 50 I=1, N
P           DO 50 J=1, N
P           P(I, J)=Q(I, J)
```

The next print out information is the initial values of R(1, 1) and Q(1, 1) followed by the calculation and print out of the noise to signal power ratio as discussed above. This print out is illustrated at the top of Sample Print Out Page 2.

```
P           PRINT 2002, R(1, 1), Q(1, 1)
P     2002  FORMAT (1H1, 8HR(1, 1)= , F12.6/1X, 8HQ(1, 1)= , F12.6)
P           PNS=R(1, 1)/Q(1, 1)
P           PRINT 2001, PNS
P     2001  FORMAT (///20X, 24HNOISE TO SIGNAL RATIO = , F10.4)
```

The next two statements in the program supply the value of the H matrix and the G matrix with H(1, 1) and G(1, 1) both equal to one (1). The H matrix is so described because only position is measureable from the plant.

```
P           H(1, 1)=1.0
P           G(1, 1)=1.0
```

The next loop in the program consists of Subroutine GANDPR, which as described in Jardine [3], provides for each sample instant, an optimum gain matrix, and the predicted values of the process and the observable

state vectors, again assuming the process is linear and the random excitation and corruptive measurement noise vectors have Gaussian distributions.

```
P          DO 1000 KK=1, 40
P          CALL GANDPR (P, Q, R, PHI, H, N, N, G)
```

The arguments of GANDPR are described below. SUBROUTINE GANDPR (P, Q, R, PHI, H, KN, KP, G)

a. P--initial covariance matrix of error

b. Q--covariance matrix of excitation (Gaussian)

c. R--covariance matrix of measurement noise

d. PHI--transition matrix of the system

e. H--matrix of defining the observable states

f. KN--Number of states in the system

g. KP--Number of observable states

h. G--Optimal gain matrix (The output of Subroutine GANDPR).

The following pages list Subroutine GANDPR and all of the Subroutines within GANDPR including a short discussion of the arguments of each of these subroutines.

```
P          SUBROUTINE GANDPR(P, Q, R, PHI, H, KN, KP, G)
P  C       THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND
P  C      1THE PREDICTED ESTIMATES FOR BOTH THE STATE VECTOR AND
P  C      2THE OBSERVABLE STATE VECTOR FOR THE NON-STATIONARY
P  C      3PROCESS FOR EACH SAMPLE PERIOD.
P          DIMENSION P(12, 12), Q(12, 12), H(12, 12), TS(12, 12), G(12, 12),
```

```
P         1R(12, 12), GT(12, 12), HT(12, 12), TEMP(12, 12), TEMP2(12, 12),
P         2TEMP1(12, 12), PHI(12, 12), PHIT(12, 12), PHIXG(12, 12)
P          CALL TRANS(H, KP, KN, HT)
P          CALL PROD (PHI, G, KN, KN, KP, PHIXG)
P          CALL PROD(PHIXG, H, KN, KP, KN, TEMP)
P          DO 108 I=1, KN
P          DO 108 J=1, KN
P      108 TEMP(I, J)= -TEMP(I, J)
P          CALL ADD(PHI, TEMP, KN, KN, TS)
P          CALL TRANS(PHI, KN, KN, PHIT)
P          CALL PROD (P, PHIT, KN, KN, KN, TEMP2)
P          CALL PROD (ITS, TEMP2, KN, KN, KN, P)
P          CALL ADD(P, Q, KN, KN, P)
P      104 CALL PROD (P, HT, KN, KN, KP, TEMP)
P          CALL PROD (H, TEMP, KP, KN, KP, TEMP2)
P          CALL ADD (TEMP2, R, KP, KP, TEMP2)
P          CALL RECIP (1 , .00000001, TEMP2, TEMP1, KER)
P          IF (KER-2) 101, 110, 101
P      110 PRINT 111
P      111 FORMAT (5HKER=2)
P      101 CALL PROD(TEMP, TEMP1, KN, KP, KP, G)
P          END
```

Subroutine Add adds two matrices which are conformable to addition.

SUBROUTINE ADD (A, B, N, M, C)

a. Adds matrix A to matrix B and results in C.

b. N and M indices as required for the addition process

```
P          SUBROUTINE ADD (A, B, N, M, C)
P          DIMENSION A(12, 12), B(12, 12), C(12, 12)
P          DO 152 I=1, N
P          DO 152 J=1, M
P      152 C(I, J) = A(I, J) + B(I, J)
P          END
```

Subroutine Trans transposes any matrix.

SUBROUTINE TRANS (A, N, M, C)

a. Transposes the N by M matrix A to a M by N matrix C

```
P          SUBROUTINE TRANS(A, N, M, C)
P          DIMENSION A(12, 12), C(12, 12)
P          DO 153 I=1, N
P          DO 153 J=1, M
P      153 C(J, I) = A(I, J)
P          END
```

Subroutine Prod multiplies two matrices which are conformable to multiplication.

SUBROUTINE PROD (A, B, N, M, L, C)

a. Multiplies any N by M matrix A by a second M by L matrix B resulting in an N by L matrix C.

```
P          SUBROUTINE PROD (A, B, N, M, L, C)
P          DIMENSION A(12, 12, B(12, 12), C(12, 12)
P          DO 151 I=1, N
P          DO 151 J=1, L
P          C(I, J) =0.
P          DO 151 K = 1, M
P      151 C(I, J) = C(I, J) + A(I, K)*B(K, J)
P          END
```

Subroutine Recip finds the inverse of a given matrix.

SUBROUTINE RECIP (N, EP, A, X, KER)

a. Finds the inverse X of a matrix A provided the determinant of A has a value greater than EP. If this value is less than EP, KER=2 indicates the error. (Inverse does not exist.) N = KP (No. of observables)

```
P          SUBROUTINE RECIP(N, EP, A, X, KER)
P          DIMENSION A(12, 12), X(12, 12)
P          DO 1 I=1, N
P          DO 1 J=1, N
P        1 X(I, J)=0. 0
P          DO 2 K=1, N
P        2 X(K, K)=1. 0
P       10 DO 34 L=1, N
P          KP=0
P          Z=0. 0
```

```
P       DO 12 K=L, N
P       IF(Z-ABSF(A(K, L)))11, 12, 12
P    11 Z=ABSF(A(K, L))
P       KP = K
P    12 CONTINUE
P       IF(L-KP)13, 20, 20
P    13 DO 14 J=L, N
P       Z-A(L, J)
P       A(L, J)=A(KP, J)
P    14 A(KP, J)=Z
P       DO 15 J=1, N
P       Z=X(L, J)
P       X(L, J)=X(KP, J)
P    15 X(KP, J)=Z
P    20 IF(ABSF(A(L, L))-EP)50, 50, 30
P    30 IF(L-N)31, 34, 34
P    31 LP1=L+1
P       DO 36 K=LP1, N
P       IF(A(K, L))32, 36, 32
P    32 RATIO=A(K, L)/A(L, L)
P       DO 33 J=LP1, N
P    33 A(K, J)=A(K. J)-RATIO*A(L, J)
P       DO 35 J=1, N
```

```
P       35 X(K, J)=X(K, J)-RATIO*X(L, J)
P       36 CONTINUE
P       34 CONTINUE
P       40 DO 43 I=1, N
P          II=N+1-I
P          DO 43 J=1, N
P          S=0.0
P          IF(II-N)41, 43, 43
P       41 IIP1=II+1
P          DO 42 K=IIP1, N
P       42 S=S+A(II, K)*X(K, J)
P       43 X(II, J)=(X(II, J)-S)/A(II, II)
P          KER=1
P          RETURN
P       50 KER=2
P          END
P          END
```

The loop containing subroutine GANDPR is set up for 40 iterations in this program. Any desired number may be used however.

The statements following the CALL GANDPR statement through statement 703 are for purposes of print out. New arrays are set up to record as vectors each term of the gain matrix and error matrix for each iteration. After all 40 iterations in this case, Print and Format statements follow the

loop to give a desirable form for the output information. **Sample Print Out Page 2** shows the results for a fourth-order system.

```
P         XG11(KK)=G(1, 1)
P         XG21(KK)=G(2, 1)
P         XG31(KK)=G(3, 1)
P         XG41(KK)=G(4, 1)
P   1000  CONTINUE
P         PRINT 700
P    700  FORMAT (//7X, 3HG11, 10X, 3HG21, 10X, 3HG31, 10X, 3HG41)
P         DO 711 KK=1, 40
P    711  PRINT 701 (XG11(KK), XG21(KK), XG31(KK), XG41(KK))
P    701  FORMAT (9F13. 5)
```

The following statements complete the main program by incrementing R(1, 1) for our so-called R(1, 1) loop, as discussed above.

For noise to signal power ratios 0 to 10 the following statements apply.

```
P         ALL=LL
P         R(1, 1)=ALL*XQ
P   1001  CONTINUE
P         END
```

For noise to signal power ratios 0 to 1.0 the following statements apply.

```
P         ALL=LL
```

R(1,1)= .00000000125604
Q(1,1)= .00000000418679

NOISE TO SIGNAL RATIO = .3000

G11	G21	G31	G41
-.12623	1.05744	27.02340	357.44429
.05189	.97639	32.59421	530.95467
.00899	.99591	33.42039	575.08506
.02019	.99081	33.53538	581.85073
.01718	.99218	33.54699	582.61782
.01799	.99181	33.54835	582.69408
.01777	.99191	33.54841	582.70121
.01783	.99188	33.54843	582.70183
.01781	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189
.01782	.99189	33.54843	582.70189

Sample print out Page 2

Appendix I

```
P          AALL=ALL/10.
P          R(1, 1)=AALL*XQ
P     1001 CONTINUE
P          END
```

The complete uninterrupted listing of Program Filtall follows.

```
      PROGRAM FILTALL

C     D1 ORDER OF SYSTEM IN I2 FORMAT
C     D2 SAMPLING INTERVAL IN F10.0 FORMAT
C     D3 F MATRIX BY ROWS IN 8F10.0 FORMAT
C     D4 D MATRIX BY COLUMN IN 8F10.0 FORMAT
C      PHI SYSTEM TRANSITION MATRIX
C      DEL DISTRIBUTION MATRIX
C      G  OPTIMUM GAIN MATRIX
C      H  OBSERVABLE MATRIX
C      P BEST ESTIMATE OF ERROR COVARIANCE MATRIX
C      Q  EXCITATION NOISE COVARIANCE MATRIX
      DIMENSION P(12,12),Q(12,12),H(12,12),R(12,12),G(12,12),PHIT(12,12)
     1,PHI(12,12),DEL(12),DELDELT(12,12),
     2XG11(40),XG21(40),XG31(40),XG41(40)
      READ 33, N
   33 FORMAT (I2)
      SIGWSQ=.25
      SIGW=SQRTF(SIGWSQ)
      DO 1001 LK=1,4
      READ 2,DT
    2 FORMAT(F10.0)
      CALL PHIDEL(PHI,DEL,N,DT)
      R(1,1)=0.0
      DO 1001 LL=1,11
      DO 699 I=1,N
      DO 699 J=1,N
      Q(I,J)=0.
      P(I,J)=0.
      H(I,J)=0.
  699 G(I,J)=0.
      DO 30 I=1,N
      DO 30 J=1,N
   30 DELDELT(I,J)=DEL(I)*DEL(J)
      DO 40 I=1,N
```

```
      DO 40 J=1,N
   40 Q(I,J)=SIGWSQ*DELDELT(I,J)
      XQ=Q(1,1)
      DO 50 I=1,N
      DO 50 J=1,N
   50 P(I,J)=Q(I,J)
      PRINT 2002, R(1,1),Q(1,1)
 2002 FORMAT (1H1,8HR(1,1)= ,F20.14/1X,8HQ(1,1)= ,F20.14)
      PNS=R(1,1)/Q(1,1)
      PRINT 2001, PNS
 2001 FORMAT (///20X,24HNOISE TO SIGNAL RATIO = ,F10.4)
      H(1,1)=1.0
      G(1,1) = 1.
      DO 1000 KK=1,40
      CALL GANDPR(P,Q,R,PHI,H,N,N,G)
      XG11(KK)=G(1,1)
      XG21(KK)=G(2,1)
      XG31(KK)=G(3,1)
      XG41(KK)=G(4,1)
 1000 CONTINUE
      PRINT 700
  700 FORMAT (//7X,3HG11,10X,3HG21,10X,3HG31,10X,3HG41)
      DO 711 KK=1,40
  711 PRINT 701 (XG11(KK),XG21(KK),XG31(KK),XG41(KK))
  701 FORMAT (9F13.5)
      ALL=LL
      AALL=ALL/10.
      R(1,1)=AALL*XQ
 1001 CONTINUE
      END

      SUBROUTINE PHIDEL(PHI,DEL,N,DT)
C     VALID ONLY FOR A CONSTANT F MATRIX
      DIMENSION F(12,12),PHI(12,12),TERM(12,12),WORM(12,12)
     1, DEL(12),DELM(12,12),TELM(12,12),DELP(12,12),D(12)
```

```
    1 FORMAT ((8F10.0))
      READ1,((F(IR,IC),IC=1,N),IR=1,N)
      READ 1 (D(I),I=1,N)
 1003 PRINT 399,DT,((F(IR,IC),IC=1,N),IR=1,N)
  399 FORMAT (1H1,8X,3HDT=,1F5.3///,7HF(I,J)=/,((8F8.2)))
      PRINT 3991 (D(I),I=1,N)
 3991 FORMAT(////5HD(I)=/(8F8.2))
      NFINAL=1
      TM=0.0
      DO 400 IR=1,N
      DO 400 IC=1,N
      TERM(IR,IC)=0.0
      WORM(IR,IC)=0.0
      TERM(IR,IR)=1.0
      TELM(IR,IC)=TERM(IR,IC)*DT
      DELP(IR,IC)=TELM(IR,IC)
      DELM(IR,IC)=0.0
      DEL(IR)=0.0
  400 PHI(IR,IC)=TERM(IR,IC)
    4 TM=1.0+TM
      DO 500   IR=1,N
      DO 500   IC=1,N
      DO 500   JN=1,N
      DELM(IR,IC)=DELM(IR,IC)-TELM(IR,JN)*F(JN,IC)*DT/(TM+1.0)
  500 WORM(IR,IC)=TERM(IR,JN)*F(JN,IC)*DT/TM+WORM(IR,IC)
      DO 401 IR=1,N
      DO 401 IC=1,N
      TERM(IR,IC)=WORM(IR,IC)
      TELM(IR,IC)=DELM(IR,IC)
      DELP(IR,IC)=DELP(IR,IC)+TELM(IR,IC)
      PHI(IR,IC)=PHI(IR,IC)+TERM(IR,IC)
      DELM(IR,IC)=0.0
  401 WORM(IR,IC)=0.0
      ABC=0.0
      DO 2I=1,N
      DO 2J=1,N
```

```
      AA=TERM(I,J)
      AB=ABSF(AA)
      IF(ABC-AB)3,3,2
    3 ABC=AB
    2 CONTINUE
      IF(0.000000005-ABC)5,5,6
    5 GO TO 4
    6 PRINT 502,((PHI(IR,IC),IC=1,N),IR=1,N)
  502 FORMAT(////9X,8HPHI(I,J)///(6E20.8))
      DO 600 I=1,N
      DO 600 K=1,N
      DO 600 J=1,N
  600 DEL(I)=DEL(I)+PHI(I,J)*DELP(J,K)*D(K)
      PRINT 503 (DEL(I),I=1,N)
  503 FORMAT(////9X,6HDEL(I)//(6E20.8)//)
      END

      SUBROUTINE GANDPR(P,Q,R,PHI,H,KN,KP,G)
C     THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND THE PREDICTED
C     ESTIMATES FOR BOTH THE STATE VECTOR AND THE OBSERVABLE STATE VECTOR
C     FOR THE NON-STATIONARY PROCESS FOR EACH SAMPLE PERIOD.
      DIMENSION P(12,12),Q(12,12),H(12,12),TS(12,12),
     1  G(12,12),R(12,12),GT(12,12),HT(12,12),TEMP(12,12),TEMP2(12,12)
     2,TEMP1(12,12),PHI(12,12),PHIT(12,12),PHIXG(12,12)
      CALL TRANS (H,KP,KN,HT)
      CALL PROD (PHI,G,KN,KN,KP,PHIXG)
      CALL PROD (PHIXG,H,KN,KP,KN,TEMP)
      DO 108 I=1,KN
      DO 108 J=1,KN
  108 TEMP(I,J)= -TEMP(I,J)
      CALL ADD (PHI,TEMP,KN,KN,TS)
      CALL TRANS(PHI,KN,KN,PHIT)
      CALL PROD (P,PHIT,KN,KN,KN,TEMP2)
      CALL PROD (TS,TEMP2,KN,KN,KN,P)
      CALL ADD(P,Q,KN,KN,P)
```

```
104 CALL PROD (P,HT,KN,KN,KP,TEMP)
    CALL PROD (H,TEMP,KP,KN,KP,TEMP2)
    CALL ADD (TEMP2,R,KP,KP,TEMP2)
    CALL RECIP (1 ,.00000001,TEMP2,TEMP1,KER)
    IF (KER-2) 101,110,101
110 PRINT 111
111 FORMAT (5HKER=2)
101 CALL PROD(TEMP,TEMP1,KN,KP,KP,G)
    END


    SUBROUTINE ADD (A,B,N,M,C)
    DIMENSION A(12,12),B(12,12),C(12,12)
    DO 152 I=1,N
    DO 152 J=1,M
152 C(I,J) = A(I,J) + B(I,J)
    END

    SUBROUTINE PROD (A,B,N,M,L,C)
    DIMENSION A(12,12),B(12,12),C(12,12)
    DO 151 I=1,N
    DO 151 J=1,L
    C(I,J) =0.
    DO 151 K = 1,M
151 C(I,J) = C(I,J) + A(I,K)*B(K,J)
    END

    SUBROUTINE TRANS(A,N,M,C)
    DIMENSION A(12,12),C(12,12)
    DO 153 I = 1,N
    DO 153 J=1,M
153 C(J,I) = A(I,J)
    END

    SUBROUTINE RECIP(N,EP,A,X,KER)
    DIMENSION A(12,12),X(12,12)
```

```
      DO 1 I=1,N
      DO 1 J=1,N
    1 X(I,J)=0.0
      DO 2 K=1,N
    2 X(K,K)=1.0
   10 DO 34 L=1,N
      KP=0
      Z=0.0
      DO 12 K=L,N
      IF(Z-ABSF(A(K,L)))11,12,12
   11 Z=ABSF(A(K,L))
      KP = K
   12 CONTINUE
      IF(L-KP)13,20,20
   13 DO 14 J=L,N
      Z=A(L,J)
      A(L,J)=A(KP,J)
   14 A(KP,J)=Z
      DO 15 J=1,N
      Z=X(L,J)
      X(L,J)=X(KP,J)
   15 X(KP,J)=Z
   20 IF(ABSF(A(L,L))-EP)50,50,30
   30 IF(L-N)31,34,34
   31 LP1=L+1
      DO 36 K=LP1,N
      IF(A(K,L))32,36,32
   32 RATIO=A(K,L)/A(L,L)
      DO 33 J=LP1,N
   33 A(K,J)=A(K,J)-RATIO*A(L,J)
      DO 35 J=1,N
   35 X(K,J)=X(K,J)-RATIO*X(L,J)
   36 CONTINUE
   34 CONTINUE
   40 DO 43 I=1,N
      II=N+1-I
```

```
      DO 43 J=1,N
      S=0.0
      IF(II-N)41,43,43
   41 IIP1=II+1
      DO 42 K=IIP1,N
   42 S=S+A(II,K)*X(K,J)
   43 X(II,J)=(X(II,J)-S)/A(II,II)
      KER=1
      RETURN
   50 KER=2
      END
      END
```

APPENDIX II

Program OPTCON minimizes a cost function by utilizing the recursive relations developed in section 4. The output of the program is the feedback gain matrix A^T which is a function of the states. The program is designed to handle all three cases as discussed in section 4. It should be noted that the Q and R of Program OPTCON are the Q and R of the cost function and they should be distinguished from the Q and R(1, 1) in program FILTALL. An unfortunate but commonly accepted notation.

Case One or the Bang-Bang Case utilizes $R = 0$, $Q = I$ therefore $P_O = I$ and Q is not used in the recursion relations for this case.

Case Two or the minimum effort control utilizes $R = 1$, $Q = I$ therefore $P_O = I$ and again Q is not used in the recursion relations.

Case Three can be used for various desired results, ie, Q and R can be given by the relations developed by Demetry [2] and discussed in section 4, or for minimum control and norm of the states, $Q = I$ and $R = 1$.

For our purposes Case Three is utilized using Q and R as developed by Demetry.

Data Cards

Data Cards are listed in the order required, if all three cases are to be used, then three sets of cards are required.

Data Card 1 N, System Order, and M, the number of stages desired, 2I10 Format.

Data Card 2 R, the Cost function constant and DT Sample Interval 2F20.14 Format.

Data Card 3 Q, the weighting matrix in the cost function 4F20.14 Format by rows.

Data Card 4 F the coefficient matrix 4F10.0 Format by rows.

Data Card 5 D the control coefficient matrix 4F10.0 Format a column.

The first series of statements in Program Optcon sets up the DO loop for the three cases, followed by reading in the data cards as listed above, followed by simple statement of PHI and DEL obtained from program FILTALL. As further indicated all of this input data is Printed Out.

```
P          DO 23 II=1,3
P          READ 8, N, M
P        8 FORMAT ((2I10))
P          READ 7, R, DT
P          READ 7, ((Q(I, J), J=1, N), I=1, N)
P        7 FORMAT (4F20.14)
P          READ 1, ((F(IR, IC), IC=1, N), IR=1, N)
P          READ 1, (D(I), I=1, N)
P        1 FORMAT ((8F10.0))
P          DO 66 I=1, N
P          DO 66 J=1, N
P       66 PHI(I, J)=0.0
P          PHI(1, 1)=1.0
```

```
P           PHI(1, 2)=0.5
P           PHI(1, 3)=.0773232
P           PHI(2, 2)=1.0
P           PHI(2, 3)=.2448333
P           PHI(3, 3)=.19204991
P           DEL(1, 1)=19.5041356
P           DEL(2, 1)=104.386352
P           DEL(3, 1)=330.525
P  C        PRINT THE DATA READ IN.
P           PRINT 999, II
P       999 FORMAT (1H1, 20X, 5HCASE , I2)
P           PRINT 43, N, M
P        43 FORMAT (/9X, 2HN=, I3, 20X, 2HM=, I3)
P           PRINT 44, R
P        44 FORMAT (/9X, 2HR=, F20.14)
P           PRINT 9, ((Q(I, J), J=1, N), I=1, N)
P         9 FORMAT (/9X, 7HQ(I, J)=/3F20.14))
P      1003 PRINT 399, DT, ((F(IR, IC), IC=1, N), IR=1, N)
P       399 FORMAT (1H1, 8X, 3HDT=, 1F5.3///, 7HF(I. J)=/, ((8F8.2)))
P           PRINT 3991 (D(I), I=1, N)
P      3991 FORMAT(////5HD(I)=/(8F8.2))
P           PRINT 11, ((PHI(I, J), J=1, N), I=1, N)
P        11 FORMAT (/9X, 9HPHI(I, J)=/(3F20.14))
```

```
P          PRINT 12, (DEL(I), I=1, N)
P       12 FORMAT (/9X, 7HDEL(I)=/(1F20.14))
```

The next statements initialize P (called P1 in the program) equal to Q.

```
P          DO 5 I=1, N
P          DO 5 J=1, N
P        5 P1(I, J)=Q(I, J)
```

The next DO loop is the main loop of the program. It calculates A^T, PSI, and P according to the recursion relationships in section 4.

```
P          DO 23 III-1, M-1
P          CALL ATRAN(AR, P1, PHI, DEL, R, N)
```

Subroutine ATRAN calculates A^T by first finding DEL^T, multiplying this into P1. This is then post multiplied by DEL to obtain most of the denominator of the recursion relationship. DEL^T*P1 is then post multiplied by PHI to obtain the numerator of the recursion relation. A DO loop follows to calculate AT(1, I). Subroutine ATRAN follows. Subroutines Prod and SUM are similar to subroutines Prod and ADD as discussed in Appendix I. Subroutines TRANCOL and TRANSQ are transpose subroutines which are self explanatory.

```
P  C       THIS SUBROUTINE CALCULATES A(K)T.
P          SUBROUTINE ATRAN (AT, P, PHI, DEL, R, N)
P          DIMENSION AT(8, 8), P(8, 8), PHI(8, 8), DEL(8, 8), DELT(8, 8),
P         1AB(8, 8), AC(8, 8), AD(8, 8)
P          CALL TRANCOL (DEL, DELT, N)
```

```
P           CALL PROD(AB, DELT, P, 1, N, N)
P           CALL PROD(AC, AB, DEL, 1, 1, N)
P           CALL PROD(AD, AB, PHI, 1, N, N)
P           DO 14 I=1, N
P        14 AT(1, I)=-AD(1, I)/(AC(1, 1)+R)
P           END
P C         THIS SUBROUTINE TRANSPOSES A SQUARE MATRIX OF MAX ORDER
P C        18 BY 8.
P           SUBROUTINE TRANSQ (A, B, N)
P           DIMENSION A(8, 8), B(8, 8)
P           DO 11 I=1, N
P           DO 11 J=1, N
P        11 B(J, I)=A(I, J)
P           END
P C         THIS SUBROUTINE MULTIPLIES ANY TWO MATRICES LIMITED TO
P C        1TWO 8 BY 8S
P C         THE ARGUMENTS ARE DEFINED AS FOLLOWS
P C         A THE PRODUCT
P C         B THE MULTIPLICAND
P C         C THE MULTIPLIER
P C         L NUMBER OF ROWS OF THE MULTIPLICAND AND PRODUCT
P C         M NUMBER OF COLUMNS OF THE MILTIPLIER AND PRODUCT
P C         N NUMBER OF COLUMNS OF THE MULTIPLICAND AND THE
```

```
P C       1NUMBER OF ROWS OF THE MULTIPLIER
P           SUBROUTINE PROD (A, B, C, L, M, N)
P           DIMENSION A(8, 8), B(8, 8), C(8, 8)
P           DO 10 I=1, L
P           DO 10 J=1, M
P           A(I, J)=0.0
P           DO 10 K=1, N
P        10 A(I, J)=A(I, J)+B(I, K)*C(K, J)
P           END
P C         THIS SUBROUTINE FINDS THE SUM OF ANY TWO SQUARE MATRICES
P C       1OF THE SAME DIMENSIONS UP TO AN 8 BY 8
P           SUBROUTINE SUM(A, B, C, N)
P           DIMENSION A(8, 8), B(8, 8), C(8, 8)
P           DO 13 I=1, N
P           DO 13 J=1, N
P        13 A(I, J)=B(I, J)+C(I, J)
P           END
P           CALL PPSI(PSI, PHI, DEL, AT, N)
```

Subroutine PPSI calculates PSI by first multiplying DEL times the just calculated A^T followed by PSI equalling PHI plus DEL*A^T.

```
P C         THIS SUBROUTINE CALCULATES PSI(K).
P           SUBROUTINE PPSI(PSI,PHI, DEL, AT, N)
P           DIMENSION PSI(8, 8), PHI(8, 8), DEL(8, 8), AT(8, 8), AB(8, 8)
```

```
P           CALL PROD(AB, DEL, AT, N, N, 1)
P           CALL SUM (PSI, PHI, AB, N)
P           END
P           CALL PP(P, PSI, P1, Q, AT, R, N, II)
```

Subroutine PP calculates P for use in updating the recursion for A^T. A^T is first transposed giving A and PSI transposed giving PSIT. PSIT is premultiplied into P1 and this relation is in turn post multiplied by PSI giving the first term of the recursion for P. $A*A^T$ is then formed and premultiplied by R. These two resulting terms are then added to Q giving the P recursion. For Case One Q and R are eliminated from the calculation. For Case Two, R is one and Q is eliminated from the calculation. In Case Three the development above applies. This is taken care of in the program with an IF statement applied to the Case indicator II of the initial DO loop in the program.

Subroutine PP follows.

```
P C         THIS SUBROUTINE CALCULATES P(K).
P           SUBROUTINE PP(P, PSI, P1, Q, AT, R, N, II)
P           DIMENSION P(8, 8), PSI(8, 8), Q(8, 8), AT(8, 8), AA(8, 8), PSIT(8, 8),
P          1AC(8, 8), AD(8, 8), AE(8, 8), P1(8, 8)
P           DO 15 I=1, N
P        15 AA(I, 1)=AT(1, I)
P           CALL TRANSQ (PSI, PSIT, N)
P           CALL PROD (AB, PSIT, P1, N, N, N)
```

```
P          CALL PROD (AC, AB, PSI, N, N, N)
P          CALL PROD (AD, AA, AT, N, N, 1)
P          DO 16 I=1, N
P          DO 16 J=1, N
P       16 AD(I, J)=R * AD(I, J)
P          IF (II-2) 30, 31, 32
P       30 DO 34 I=1, N
P          DO34 J=1, N
P       34 P(I, J)=AC(I, J)
P          GO TO 33
P       31 CALL SUM (P, AC, AD, N)
P          GO TO 33
P       32 CALL SUM(AE, AC, Q, N)
P          CALL SUM (P, AE, AD, N)
P       33 CONTINUE
P          END
P C        THIS SUBROUTINE TRANSPOSES A COLUMN MATRIX HAVING A
P C       1MAXIMUM OF EIGHT ELEMENTS
P          SUBROUTINE TRANCOL (A, B, N)
P          DIMENSION A(8, 8), B(8, 8)
P          DO 12 I=1, N
P       12 B(1, I)=A(I, 1)
P          END
```

The remainder of the program is concerned with printing out the results, P and A^T and updating P1 for use in the next iteration. A sample of the print out received from Program Optcon appears as Sample Print Out Pages 1 and 2 for Appendix II.

```
P          CALL PP(P, PSI, P1, Q, AT, R, N, II)
P          PRINT 19. III
P       19 FORMAT (/2HU(, 12, 12H)=AT(J)*X(J))
P          DO 20 IIII-1, N
P          PRINT 20, IIII, AT(1, IIII)
P       20 FORMAT (/9X, 3HAT(, 12, 2H)=, 1F20.10)
P          PRINT 21, ((P(I, J), J=1, N), I=1, N)
P       21 FORMAT (/6HP(I, J)/(3E20.5))
P          DO 29 J=1, N
P          DO 29 K=1, N
P       29 P1(J, K)=P(J, K)
P       23 CONTINUE
P          END
```

The Sample Print Out Pages are followed by a complete uninterrupted listing of Program OPTCON.

CASE 3

N= 3 M= 20

R= .0000000495000

Q(I,J)=

1.00000000000000	.00000000000000	.00000000000000

Q(I,J)=

.00000000000000	.00249999999994	.00000000000000

Q(I,J)=

.00000000000000	.00000000000000	.00006869999999

Sample print out Page 1

Appendix II

```
DT= .050

F(I,J)=
      .00      1.00      .00      .00      .00     1.00      .00      .00
   -3.30

D(I)=
      .00       .00 1350.00

      PHI(I,J)=
   1.00000000000000      .04999999999927     .00118399999999
    .00000000000000     1.00000000000000     .04609999999957
    .00000000000000      .00000000000000     .84699999999430

      DEL(I)=
    .02700209999966
   1.59839309999370
  62.22530299983919

U( 1)=AT(J)*X(J)

      AT( 1)=          -.0988646416

      AT( 2)=          -.0195739979

      AT( 3)=          -.0140486978

P(I,J)
        .19973E+01          .49471E-01          .80466E-03
        .49471E-01          .73954E-02          .99344E-04
        .80466E-03          .99344E-04          .70796E-04

U( 2)=AT(J)*X(J)

      AT( 1)=          -.5699646281

      AT( 2)=          -.0887031064

      AT( 3)=          -.0155427371

P(I,J)
        .28930E+01          .13310E+00          .24815E-02
        .13310E+00          .17308E-01          .30715E-03
        .24815E-02          .30715E-03          .75183E-04

U( 3)=AT(J)*X(J)

      AT( 1)=         -1.0643207535

      AT( 2)=          -.1736192142

      AT( 3)=          -.0174117600

P(I,J)
        .34191E+01          .20044E+00          .39100E-02
        .20044E+00          .27740E-01          .53410E-03
        .39100E-02          .53410E-03          .80135E-04
```

```
      PROGRAM OPTCON
      DIMENSION PHI(8,8),PSI(8,8),P(8,8),P1(8,8),DEL(8,8),
     1 AT(8,8),Q(8,8),F(8,8),D(8)

C THIS PROGRAM UTILIZES A COST FUNCTION, J(N)=MINIMUM(SUM X(N)T*Q*X(N)+
C SUM R*U(N-1)**2).  AN UNLIMITED NUMBER OF ITERATIONS MAY BE MADE AT
C A COMPUTATION RATE OF 2000 PER MINUTE AFTER THE PROGRAM HAS BEEN
C COMPILED.  THE OUTPUT OF THIS PROGRAM IS THE FEEDBACK GAIN MATRIX,
C A TRANSPOSE.  THE FOLLOWING RECURSIVE EQUATIONS WERE DERIVED USING
C DYNAMIC PROGRAMMING,

C     A(K)T=-(DELT*P(K-1)*PHI)/(DELT*P(K-1)*DEL+R)

C     PSI(K)=PHI+DEL*A(K)T,    PSI(0)=0

C     P(K)=PSI(K)T*P(K-1)*PSI(K)+Q+R*A(K)*A(K)T,   P(0)=Q

C     DATA CARDS ARE LISTED IN THE ORDER REQUIRED,IF  ALL THREE CASES
C     ARE TO BE USED, THREE SETS OF CARDS ARE REQUIRED
C     DATA CARD 1---N (SYSTEM ORDER) AND M (STAGES) 2I10 FORMAT
C     DATA CARD 2 R(CONSTANT IN COST FUNCTION) AND DT (SAMPLE INTERVAL)
C     4F20.14 FORMAT
C     DATA CARD 3   Q (WEIGHTING MATRIX USED IN THE COST FUNCTION)
C     4F20.14 FORMAT BY ROWS
C     F  (THE COEFFICIENT MATRIX) 4F10.0 FORMAT BY ROWS
C     D (THE CONTROL COEFFICIENT MATRIX) 4F10. FORMAT A COLUMN
C     CALCULATE A(K)T, PSI(K),AND P(K).
C     CALL IN DATA AND INITIALIZE

      DO 23 II=1,3
      READ 8,N,M
    8 FORMAT ((2I10))
      READ 7,R,DT
      READ 7,((Q(I,J),J=1,N),I=1,N)
    7 FORMAT (4F20.14)
      READ 1,((F(IR,IC),IC=1,N),IR=1,N)
```

```
      READ 1,(D(I),I=1,N)
    1 FORMAT ((8F10.0))
      DO 66 I=1, N
      DO 66 J=1, N
   66 PHI(I,J)=0.0
      PHI(1,1)=1.0
      PHI(1,2)=.05
      PHI(1,3)=.001247
      PHI(1,4)=.00002016
      PHI(2,2)=1.0
      PHI(2,3)=.049766
      PHI(2,4)=.0011962
      PHI(3,3)=.986123
      PHI(3,4)=.0467515
      PHI(4,3)=-.54231813
      PHI(4,4)=.86830965
      DEL(1,1)=.0001294087
      DEL(2,1)=.01283806
      DEL(3,1)=.61008364
      DEL(4,1)=23.843297

C     PRINT THE DATA READ IN.

      PRINT 999,II
  999 FORMAT (1H1,20X,5HCASE ,I2)
      PRINT 43,N,M
   43 FORMAT (/9X,2HN=,I3,20X,2HM=,I3)
      PRINT 44,R
   44 FORMAT (/9X,2HR=,F20.14)
      PRINT 9,((Q(I,J),J=1,N),I=1,N)
    9 FORMAT (/9X,7HQ(I,J)=/3F20.14))
 1003 PRINT 399,DT,((F(IR,IC),IC=1,N),IR=1,N)
  399 FORMAT (1H1,8X,3HDT=,1F5.3///,7HF(I,J)=/,((8F8.2)))
      PRINT 3991 (D(I),I=1,N)
 3991 FORMAT(////5HD(I)=/(8F8.2))
      PRINT 11,((PHI(I,J),J=1,N),I=1,N)
```

```
   11 FORMAT (/9X,9HPHI(I,J)=/(3F20.14))
      PRINT 12,(DEL(I),I=1,N)
   12 FORMAT (/9X,7HDEL(I)=/(1F20.14))
      DO 5 I=1,N
      DO 5 J=1,N
    5 P1(I,J)=Q(I,J)
      DO 23 III=1,M-1
      CALL ATRAN(AT, P1, PHI, DEL, R, N)
      CALL PPSI(PSI,PHI,DEL,AT,N)
      CALL PP(P,PSI,P1,Q,AT,R,N,II)
      PRINT 19,III
   19 FORMAT (/2HU(,I2,12H)=AT(J)*X(J))
      DO 20 IIII=1,N
      PRINT 20, IIII,AT(1,IIII)
   20 FORMAT (/9X,3HAT(,I2,2H)=,1F20.10)
      PRINT 21,  ((P(I,J),J=1,N),I=1,N)
   21 FORMAT (/6HP(I,J)/(3E20.5))
      DO 29 J=1,N
      DO 29 K=1,N
   29 P1(J,K)=P(J,K)
   23 CONTINUE
      END

C     THIS SUBROUTINE CALCULATES A(K)T.
      SUBROUTINE ATRAN (AT,P,PHI,DEL,R,N)
      DIMENSION AT(8,8),P(8,8),PHI(8,8),DEL(8,8),
     1DELT(8,8),AB(8,8),AC(8,8),AD(8,8)
      CALL TRANCOL (DEL,DELT,N)
      CALL PROD(AB,DELT,P,1,N,N)
      CALL PROD(AC,AB,DEL,1,1,N)
      CALL PROD(AD,AB,PHI,1,N,N)
      DO 14 I=1,N
   14 AT(1,I)=-AD(1,I)/(AC(1,1)+R)
      END
C     THIS SUBROUTINE CALCULATES PSI(K).
```

```
      SUBROUTINE PPSI(PSI,PHI,DEL,AT,N)
      DIMENSION PSI(8,8),PHI(8,8),DEL(8,8),AT(8,8),AB(8,8)
      CALL PROD(AB,DEL,AT,N,N,1)
      CALL SUM (PSI,PHI,AB,N)
      END
C     THIS SUBROUTINE CALCULATES P(K).

      SUBROUTINE PP(P,PSI,P1,Q,AT,R,N,II)
      DIMENSION P(8,8),P1(8,8),PSI(8,8),Q(8,8),AT(8,8),
     1AA(8,8),PSIT(8,8),AC(8,8),AD(8,8),AE(8,8)
      DO 15 I=1,N
   15 AA(I,1)=AT(1,I)
      CALL TRANSQ (PSI,PSIT,N)
      CALL PROD (AB,PSIT,P1,N,N,N)
      CALL PROD (AC,AB,PSI,N,N,N)
      CALL PROD (AD,AA,AT,N,N,1)
      DO 16 I=1,N
      DO 16 J=1,N
   16 AD(I,J)=R * AD(I,J)
      IF (II-2) 30,31,32
   30 DO 34 I=1,N
      DO34 J=1,N
   34 P(I,J)=AC(I,J)
      GO TO 33
   31 CALL SUM (P,AC,AD,N)
      GO TO 33
   32 CALL SUM(AE,AC,Q,N)
      CALL SUM (P,AE,AD,N)
   33 CONTINUE
      END
C     THIS SUBROUTINE TRANSPOSES A COLUMN MATRIX HAVING A MAXIMUM OF
C     EIGHT ELEMENTS
      SUBROUTINE TRANCOL (A,B,N)
      DIMENSION   A(8,8),B(8,8)
      DO 12 I=1,N
   12 B(1,I)=A(I,1)
```

```
      END
C     THIS SUBROUTINE TRANSPOSES A SQUARE MATRIX OF MAX ORDER 8 BY 8.
      SUBROUTINE TRANSQ (A,B,N)
      DIMENSION A(8,8),B(8,8)
      DO 11 I=1,N
      DO 11 J=1,N
   11 B(J,I)=A(I,J)
      END
C     THIS SUBROUTINE MULTIPLIES ANY TWO MATRICES LIMITED TO TWO 8 BY 8S
C     THE ARGUMENTS ARE DEFINED AS FOLLOWS
C     A THE PRODUCT
C     B THE MULTIPLICAND
C     C THE MULTIPLIER
C     L NUMBER OF ROWS OF THE MULTIPLICAND AND PRODUCT
C     M NUMBER OF COLUMNS OF THE MULTIPLIER AND PRODUCT
C     N NUMBER OF COLUMNS OF THE MULTIPLICAND AND THE NUMBER OF ROWS OF
C     THE MULTIPLIER
      SUBROUTINE PROD (A,B,C,L,M,N)
      DIMENSION A(8,8),B(8,8),C(8,8)
      DO 10 I=1,L
      DO 10 J=1,M
      A(I,J)=0.0
      DO 10 K=1,N
   10 A(I,J)=A(I,J)+B(I,K)*C(K,J)
      END
C     THIS SUBROUTINE FINDS THE SUM OF ANY TWO SQUARE MATRICES OF THE
C     SAME DIMENSIONS UP TO AN 8 BY 8
      SUBROUTINE SUM(A,B,C,N)
      DIMENSION A(8,8),B(8,8),C(8,8)
      DO 13 I=1,N
      DO 13 J=1,N
   13 A(I,J)=B(I,J)+C(I,J)
      END
      END
```

APPENDIX III

Program CLOSE utilizes the recursion relationships as shown in section 7 to close the loop on the control problem utilizing the results of program FILTALL, the gain matrix G, and the results of Program OPTCON, the feedback gain matrix A transpose. These two gain matrices are computed on the basis of the desired response, and the statistical properties of the anticipated random disturbance and measurement noise. The system may be driven either by initial conditions or by step or ramp inputs, or any combination of these. The program solves the following equations:

Y(K) = H*X(K)

Z(K) = Y(K) + V(K)

XS(K) = (I-GH)*PHI*XS(K-1) + (I-GH)*DEL*AT*(XS(K-1)-DINP(K-1)

+ G*Z(K)

X(K+1) = PHI*X(K) + DEL*AT*(XS(K)-DINP(K)) + DEL*W(K)

where V is the measurement noise, W is the random disturbance, and DINP is the deterministic input.

No data cards are required for this program as all the required information is entered directly. This is a matter of preference as data cards could be utilized to read in the required data. G, H, PHI, DEL, SIGW, DT, SIGV (1), KN, and KP are found in Program FILTALL, while A transpose is found from OPTCON.

The subroutines found in the program, PROD and ADD are similar to those utilized in Program FILTALL and OPTCON and described in Appendix I.

Deterministic inputs must be designed in the proper manner to obtain the desired results. These require a certain amount of calculation when the system includes a zero and/or ramp inputs are utilized.

Program CLOSE is completely listed below with appropriate comment cards throughout the program to ensure clarity.

```
      PROGRAM CLOSE
      DIMENSION X(12,12),XS(12,12),SIGV(12),Y(12,12),Z(12,12),PHI(12,12)
     1,DEL(12,12),H(12,12),AT(12,12),G(12,12),TEMP1(12,12),TEMP2(12,12),
     2TEMP3(12,12),TEMP4(12,12),TELP(12,12),TELP1(12,12),TELP2(12,12),
     3V(12,12),DINP(12,12),
     1YY(200)
C
C     THIS PROGRAM CLOSES THE LOOP ON THE OPTIMUM FILTER-CONTROLLER
C     PROBLEM. IT ASSUMES THAT STABLE CONTROLLER AND FILTER GAIN MATRICES
C     HAVE BEEN COMPUTED ON THE BASIS OF DESIRED RESPONSE AND THE
C     STATISTICAL PROPERTIES OF THE ANTICIPATED RANDOM DISTURBANCE AND
C     MEASUREMENT NOISE. THE SYSTEM MAY BE DRIVEN EITHER BY INITIAL
C     CONDITIONS OR BY STEP OR RAMP INPUTS,OR BY ANY COMBINATION OF
C     THESE.
C
C     THE PROGRAM SOLVES THE FOLLOWING EQUATIONS
C         Y(K)=H*X(K)
C         Z(K)=Y(K)+V(K)
C     XS(K)=(I-GH)PHI*XS(K-1)+(I-GH)*DEL*AT(XS(K-1)-DINP(K-1))+G*Z(K)
C     X(K+1)=PHI*X(K)+DEL*W(K)+DEL*AT*(XS(K)-DINP(K))
C     WHERE V IS MEASUREMENT NOISE,W IS THE RANDOM DISTURBANCE,AND
C     DINP IS THE DETERMINISTIC INPUT
C
C     INITIALIZE SYSTEM AND FILTER STATE VARIABLES. LATTER ARE GUESSES.

      X(1,1)=0.0
      X(2,1)=0.0
      X(3,1)=0.0
      X(4,1)=0.0
      XS(1,1)=0.0
      XS(2,1)=0.0
      XS(3,1)=0.0
      XS(4,1)=0.0

C     INPUT SOME CONSTANTS, AS NOTED
```

```
      DT=.05
C     KN IS SYSTEM ORDER. KP IS THE NUMBER OF OBSERVABLES.
      KN=4
      KP=1
C     FMULT IS INVOLVED IN UPDATING THE DETERMINISTIC INPUT
      FMULT=0.0
      STEP= 1./.0455
      RAMP=0.0
      DINP(1,1)=0.0
      DINP(2,1)=0.0
      DINP(3,1)=0.0
      DINP(4,1)=0.0
C     SETTING DINP=0.0 AT THIS POINT INSURES THAT NO DETERMINISTIC
C     INPUTS EXIST PRIOR TO TIME = ZERO
      DO 65 I=1,KN
   65 SIGV(I)=0.0
      SIGV(1,1)=SQRTF(.00000000041868)
      SIGW=.5
C     CARDS ABOVE PROVIDE FOR SETTING DESIRED VARIANCES IN W AND V.
C     SIGW IS THE STANDARD DEVIATION OF W. SIMILARLY FOR SIGV.
C     INPUT PHI,DEL,G,H,AND AT MATRICES
      DO 66 I=1,KN
      DO 66 J=1,KN
   66 PHI(I,J)=0.0
      PHI(1,1)=1.0
      PHI(1,2)=.05
      PHI(1,3)=.001247
```

```
      PHI(1,4)=.00002016
      PHI(2,2)=1.0
      PHI(2,3)=.049766
      PHI(2,4)=.0011962
      PHI(3,3)=.986123
      PHI(3,4)=.0467515
      PHI(4,3)=-.54231813
      PHI(4,4)=.86830965

      DEL(1,1)=.0001294087
      DEL(2,1)=.01283806
      DEL(3,1)=.61008364
      DEL(4,1)=23.843297

      DO 67 I=1,KN
      DO 67 J=1,KN
   67 G(I,J)=0.0
      G(1,1)=.01782
      G(2,1)=.99189
      G(3,1)=33.54936
      G(4,1)=582.73566

      DO 68 I=1,KN
      DO 68 J=1,KN
   68 H(I,J)=0.0
      H(1,1)=.455
      H(1,2)=1.0
      H(1,3)=0.0

      AT(1,1)=-1.8320888409
      AT(1,2)=-4.4257382891
      AT(1,3)=-.6917466321
      AT(1,4)=-.0517540388

C     THE FOLLOWING CARD INITIALIZES THE RANDOM NUMBER GENERATOR
```

```
      NUNIF=1220703125
      PRINT 800
  800 FORMAT (1H1,6X,4HX(1),10X,4HX(2),10X,4HX(3),10X,4HX(4),10X,5HXS(1)
     1,9X,5HXS(2),9X,5HXS(3),9X,5HXS(4))
   70 DO 900 KK=1,200
      CALL PROD(H,X,KP,KN,1,Y)
      DO 10 I=1,KP
      CALL RNDEV(NUNIF,DEV)
   10 V(I,1)=SIGV(I)*DEV
      CALL ADD(Y,V,KP,1,Z)
      CALL PROD(PHI,XS,KN,KN,1,TEMP1)
      CALL PROD(G,H,KN,KP,KN,TEMP2)
      DO 11 I=1,KN
      DO 11 J=1,KN
   11 TEMP2(I,J)=-TEMP2(I,J)
      CALL PROD(TEMP2,TEMP1,KN,KN,1,TEMP3)
      CALL ADD(TEMP1,TEMP3,KN,1,TEMP3)
      CALL ADD(XS,DINP,KN,1,TELP)
      CALL PROD(AT,TELP,1,KN,1,TEMP1)
      CALL PROD(DEL,TEMP1,KN,1,1,TELP)
      CALL PROD(TEMP2,TELP,KN,KN,1,TELP1)
      CALL ADD(TELP,TELP1,KN,1,TELP1)
      CALL PROD(G,Z,KN,KP,1,TELP2)
      CALL ADD(TEMP3,TELP1,KN,1,XS)
      CALL ADD(XS,TELP2,KN,1,XS)
      DINP(1,1)=-(FMULT*RAMP+STEP)
      DINP(2,1)=-RAMP/DT
      FMULT=FMULT+1.0
      CALL RNDEV(NUNIF,DEV)
      W=SIGW*DEV
      PRINT 801, X(1,1),X(2,1),X(3,1),X(4,1),XS(1,1),XS(2,1),XS(3,1),
     1XS(4,1)
  801 FORMAT (8F14.6)
      YY(KK)=.455*X(1)+X(2)
      CALL PROD(PHI,X,KN,KN,1,TEMP1)
      DO 803 I=1,KN
```

```
 803 TEMP2(I,1)=W*DEL(I,1)
     CALL ADD(XS,DINP,KN,1,TELP)
     CALL PROD(AT,TELP,1,KN,1,TELP1)
     CALL PROD(DEL,TELP1,KN,1,1,TELP2)
     CALL ADD(TEMP1,TEMP2,KN,1,X)
     CALL ADD(X,TELP2,KN,1,X)
 900 CONTINUE
     PRINT 1886
1886 FORMAT(1H1,14X,1HY)
     DO 1885 KK=1,200
1885 PRINT 1887,YY(KK)
1887 FORMAT (//F20.6)
     END


     SUBROUTINE PROD (A,B,N,M,L,C)
     DIMENSION A(12,12),B(12,12),C(12,12)
     DO 151 I=1,N
     DO 151 J=1,L
     C(I,J)=0.
     DO 151 K=1,M
 151 C(I,J)=C(I,J)+A(I,K)*B(K,J)
     END

     SUBROUTINE ADD (A,B,N,M,C)
     DIMENSION A(12,12),B(12,12),C(12,12)
     DO 152 I=1,N
     DO 152 J=1,M
 152 C(I,J)=A(I,J)+B(I,J)
     END
     END
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library U. S. Naval Postgraduate School Monterey, California	2
3.	Bureau of Naval Weapons Department of the Navy Washington, D. C.	1
4.	Prof Robert D. Strum (Thesis Advisor) Department of Electrical Engineering U. S. Naval Postgraduate School Monterey, California	10
5.	LT John Coleman McMichael, USN 1233 Drexel Ave. Drexel Hill, Pa.	2
6.	LT George Christ Skezas, USN 562 West Third Street Huron, South Dakota	2

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* U. S. Naval Postgraduate School	**2a. REPORT SECURITY CLASSIFICATION** Unclassified **2b. GROUP**

3. REPORT TITLE

Case Studies of Optimum Filter-Controller Design in Sampled Data Systems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Masters Thesis May 1966

5. AUTHOR(S) *(Last name, first name, initial)*

McMichael, John C. Skezas, George C.

6. REPORT DATE	**7a. TOTAL NO. OF PAGES**	**7b. NO. OF REFS**
May 1966	96	6

8a. CONTRACT OR GRANT NO.	**9a. ORIGINATOR'S REPORT NUMBER(S)**
b. PROJECT NO.	
c.	**9b. OTHER REPORT NO(S)** *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

[redacted] ~~may obtain copies of this report from DDC~~ #1 [illegible] 12/22/69

11. SUPPLEMENTARY NOTES	**12. SPONSORING MILITARY ACTIVITY** Bureau of Naval Weapons

13. ABSTRACT In this paper an investigation is made of the problem of estimating and predicting the states of a linear, discrete, time-invariant, dynamic process which is excited by Gaussian noise and where the observable states are disturbed by Gaussian measurement noise. The concepts of optimum filter design, originally developed by R. E. Kalman, are utilized. Also we have closed the loop on two illustrative examples by determining the optimal control for the plant as a function of the plant's state variables. Here the concepts of a cost performance index and dynamic programming (the latter originally developed by R. Bellman) are employed.

The CDC 1604 digital computer, using Fortran 60 programming is utilized in the solution of the optimum filter-controller design.

DD FORM 1 JAN 64 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Control Automatic Control Optimum control Sampled-Data control State Estimation Discrete systems Optimum filter						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.